Alignment of Local and Global Features from Multiple Layers of Convolutional Neural Network for Image Classification

Fernando Pereira dos Santos and Moacir Antonelli Ponti
Institute of Mathematical and Computer Sciences (ICMC)
University of São Paulo (USP), São Carlos, SP, Brazil
Email: {fernando_persan,ponti}@usp.br

***Abstract*— Convolutional networks have been extensively applied to obtain features spaces for classification tasks. Although those achieve high accuracy in many scenarios, typically only the top layers of the network are explored. Hence, a relevant question arises from this fact: are initial layers useful in terms of discriminative ability? In this paper, we leverage the complementary description offered by such first layers. Our method consists of features extraction in multiple layers, followed by feature selection, fusion of feature maps from the different layers, and space alignment. Through an extensive experimentation with different datasets and studying different training strategies, our results show that local information, coming from the first layers, may significantly improve the classification performance when merged with a global descriptor extracted from a top layer of the network. We report different methods for reducing the dimensionality of the local descriptors, and guidelines on how to align them so that to perform fusion. Our study encourages future studies on combining feature maps from multiple layers, which may be relevant in particular for transfer learning scenarios.**

I. Introduction

Convolutional Neural Networks (CNNs) have been widely used for image classification tasks in several applications, such as relevance sampling [1], plant identification [2], remote sensing scene [3], and medical diagnosis [4], [5]. CNNs are composed of a hierarchical and sequential architecture, in which the output of one layer is the input of the next layer. In this composition, the coupled layers may have different functions, from filtering (convolution) and dimensionality reduction (pooling) to normalization [6]. One of the great advantages of the employability of the CNN is its abstraction capacity, which provides different descriptors for low-level features (shapes, edges, and colors) and high-level features (texture and semantics) [7]. In this scenario, a valuable resource is to consider a pre-trained CNN, e.g using ImageNet [8]. Hence, since the network parameters are already weighted, each layer provides a different feature map. This prior training is important because, in practice, a dataset rarely have enough examples to provide a convergence during the network training [9].

Another possibility is to fine-tune the network with a dataset similar enough to provide additional semantics to parameters already weighted in the network [10]. Consequently, feature learning methods, such CNNs, are more advantageous than handcrafted descriptors due to the feature space generalization. To fine-tune a CNN it is necessary to change its prediction layer (the last one) to contain the same number of classes from the dataset applied. In sequence, the network is retrained for a number of epochs [6]. Regardless of the approach chosen, the attributes extracted can be transformed by different methods to reduce the dimensionality [5], [11], concatenation of feature maps [12], [13], and alignment of the data distributions for transfer learning tasks [14].

Despite the great applicability of CNNs in classification tasks, the great percentual of methodologies focus on selecting one of the last layers (global descriptors) of the network to provide the feature spaces [5], [14]–[17]. This occurs due to the hierarchical structure and the loss functions applied in the network, in which the convergence of the model propagates from the last layer to the first one [6]. Therefore, the last layers are results of transformation and combination of the attributes existing in the previous layers. Consequently, these layers (the last ones) incorporate descriptors capable of distinguishing classes, in addition to having dimensionality smaller than the first layers (local descriptors). This property is effect of the receptive fields propagation from a region located in a previous layer and that will result in a specific attribute in a subsequent layer [18]. However, whenever there is dimensionality reduction there is loss of information. Also, we can not assume that only the last ones as good descriptors while first layers do not provide representativeness for classification. On the contrary, because they offer a low-level description of shapes, borders, and colors, they may play an important role in the task [10]. This is the main motivation for this paper.

Using a pre-trained ResNet-50 [19], we extracted features from the pre-prediction layer (as global descriptor) and explored some of the first layers (as local descriptors) to merge them in a single feature map (as fusion descriptor). Due to the larger amount of features coming from the first layers, we selected the attributes by three different methods. This process is performed for the dataset which will be the training set of the classifier (source) and for the test set (target). Hence, with the multi-layer attributes, from source and target, we aimed to reduce the discrepancy and increase the correlation between both data distributions (source and target) [20] using Transfer Component Analysis (TCA) [21]. As result, the new features spaces are applied to Support Vector Machine (SVM)

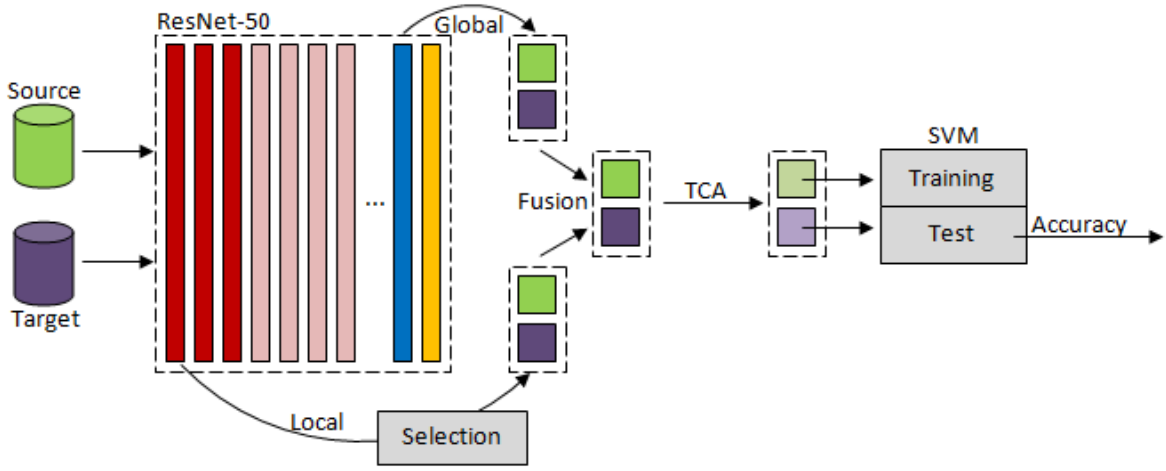

Fig. 1. Feature extraction and manifold alignment structure. Considering two datasets, source and target, each one are passed on to ResNet-50 for feature extraction. Initially, an initial layer (the red ones) provides local attributes (shape, border, and color) and the pre-prediction layer (the blue one) provides global attributes (texture and semantics). Consequently, with these two feature spaces and by means of map concatenation one obtains the fusion of the multiple layer activations for both datasets, source (green) and target (purple). After that, TCA transforms both resulting fusion space and assigning the source features to train the classifier (SVM) and the target features to be classified. Our experiments were performed in two scenarios: (i) using ResNet-50 pre-trained with ImageNet; and (ii) performing fine-tuning with the source dataset. In both cases the process of obtaining the features is the same. For comparison purposes, local and global attributes are also transformed and classified. Local attributes undergo feature selection prior to fusion.

for training and classification, as illustrated on Fig. 1.

Consequently, our contribution includes: (i) a sequential novel that aggregates multi-layer features fusion from a convolutional network, applying fine-tuning or using a pre-trained model, and manifold alignment of the data distribution for image classification; (ii) practical evidence that multi-layer features fusion provides better performance for transfer learning in low-level appearance datasets; and (iii) extensive experimentation in different scenarios of images.

II. State-the-art context

To obtain features spaces using CNNs it is common to use the end-layers of the architecture alone, due to they are receptive fields from previous ones [5], [6], [18]. However, the multi-layer feature fusion methodology has spread to several areas of image processing, including classification [22], [23], segmentation [24], and even for edge detection [25]. The feature fusion can be performed using distinct extractors, such as a CNN layer for global descriptor and handcrafted methods to describe low-level features for edges and shapes [26]. Another possible approach is to combine layers from different CNNs, however, in this case it usually applies only the high-level layers [27].

Considering the initial and mid-level layers of CNNs, Yu et al. [23] present the importance of these units showing that they only need to be extracted, represented, and fused properly. However, only the mid-level layers were exploited in their method for network flow unification to compose the pre-prediction layer for feature extraction. In the same context, building new branches within CNN from a low-level layer can generate special combinations of features to provide multiple outputs [28], [29]. In a more related manner to this study, Zheng et al. [22] evidenced the training and image classification of different CNNs by comparing the low-level attributes (local features) in relation to the high-level attributes (global features). Based on their experiments, using geometric shapes datasets, it is noted that the low-level features spaces stand out when the data distribution between the datasets (source and target) are more similar. This evidence is proven in our results by using different scenarios in which images range from fruits (where shapes and colors are most relevant descriptors) to photos (where scenes have different objects).

In addition to the approaches that only extract features in a selected layer, the attributes can undergo post-processing as an alignment of the data distribution. This methodology was explored by We et al. [30]. In this study the features were extracted in the last convolutional layer of the VGG [18] to describe the local attributes followed by the alignment. One of the foundations underlying their study is that low-level layers are more general than top-layers, even for different image domains. Consequently, our method relates the descriptive ability of multi-layer attributes followed by the data distribution alignment to favor image classification, especially in transfer learning tasks.

III. Alignment of Multi-layer features

Residual Networks [19] are CNN models that introduced the concept of residual blocks that allow training networks with a larger number of layers. They aim to preserve features from the input vector before its transformation, adding it to the output after some convolutions of delimited block, as illustrated in Fig. 2. Its version with 50 layers, ResNet-50 is widely used for transfer learning. Another interesting property of ResNets is the absence of intermediate representations learned on dense layers: an average pooling is used after the last residual block before the prediction/output layer [19].

Using the ResNet-50 we extracted features from two layers simultaneously: (i) activation maps from the first layers of the network that encode local features, and referred to as **local descriptor**, and (ii) a **global descriptor** that is the activation map obtained after the Average Pooling layer, just before prediction. This layer is commonly selected to offer representativeness of the images to be classified [5], [27], [31], providing 2048 attributes. As a local descriptor, we select the outputs of the first three residual blocks contained in the initial part of the network, individually, providing 774400 attributes each feature map. Consequently, three scenarios are presented for the multi-layer features fusion: global descriptor with each local descriptor. The fusion process is based only on the concatenation of attributes. The selection of layers in different stages of the network provides an evolution view of the features transformation and their combination.

A. Local features selection

Due to the large number of parameters from the residual blocks, 774400 in total, feature selection methods were applied to choose attributes for the fusion. Three methodologies were adopted, see Fig. 3, aiming a comparative analysis of performances: Principal Componentes Analysis (PCA); Flatten Pooling; and Pooling 2D.

a) ***PCA****:* is a classic dimensionality reduction technique in which the principal attributes are selected. In this context, for each feature its variance is calculated and those with greater discriminative capacity are chosen. PCA has a restriction on the number of attributes: the maximum number of components desired is the minimum value between samples and attributes. In our approach, PCA is applied only to the source dataset for definition of components. In sequence, the chosen components are applied to the target dataset.

b) ***Flatten Pooling****:* is a simple attribute selection method in which the feature maps are fully converted from matrix to vector without any spatial relationship. After that, a value x will split the vector into small symmetric segments. For each of these segments the average is calculated. Then, the final attributes are compounds by these averages.

c) ***Pooling 2D****:* considers a square region in the attribute space, calculating the average of this region. Consequently, location dependency inside of same feature map is considered in this method.

The layers that constitute the output of the residual blocks have the same shape: 256 maps of size 55 $\times$ 55, resulting in 774400 attributes. Empirically, we adopted 256 components for PCA and $x = 100$ for the Flatten Pooling, which provides 7744 initial attributes. For Pooling 2D, the adopted region was 55 $\times$ 55, where each map only provides one attribute. However, due to the PCA constraint, some datasets do not have 256 examples in the test set. For this specific case were determined 128 components. This variation in the amount of components is suppressed with the TCA that defines the real amount of attributes to be used in the classification.

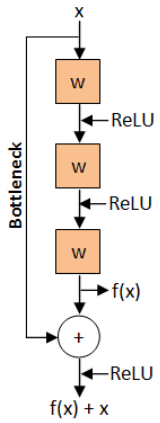

Fig. 2. Residual blocks perform sum of an input x with a data transformation $f(x)$, where w represents a convolutional layer. Bottleneck proposes to compress the input depth by a reduced number of filters and restore it before the sum. After each convolutional w there is a batch normalization layer with activation by ReLU.

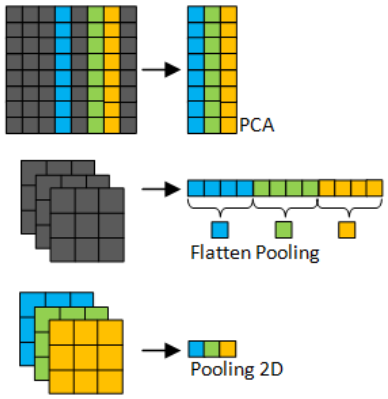

Fig. 3. Feature selection methods: PCA considers the features of all images to find the principal components. In contrast, the selection of features by Flatten Pooling and Pooling 2D is performed in an invidual manner.

B. Transfer Components Analysis (TCA)

TCA [21] attempts to learn a common set of underlying transferable components from both domains in which the difference in data distribution can be dramatically reduced with properties preserved in subspace projection. Assuming 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 Φ by two pre-definitions: (i) the distance between the two distributions generated $P(\Phi(X_s))$ and $P(\Phi(X_t))$ is small enough; and (ii) transformation Φ preserves important properties of X_s and X_t. With these statements, Φ ensures $P(Y_s|\Phi(X_s)) \approx P(Y_t|\Phi(X_t))$. In this new transformed space a classifier should be trained using the space from $\Phi(X_s)$ and applied in the target feature space $\Phi(X_t)$ for predictions. Due to its theoretical basis being derived from the PCA, for the feature space transformation to be performed it is also necessary to define the amount of attributes desired for the output [14].

IV. Experiments

A. Datasets

In transfer learning tasks it is essential that the datasets employed be similar in their data distribution and with equivalent classes [14]. Due to this factor, our results were obtained considering a dataset as source (the larger one) and another similar dataset as the target (the smaller one). In our experiments we applied four sets of different domains, considering fruits, objects, skin lesions, and photos. This diversity is extremely important to emphasize the contribution of this study, due to variation of styles, scene composition, and degree of task difficulty. These sets are described in the following:

a) ***Fruits****:* Considering the fruit domain, see Fig. 4, Fruits-360 [32] is a dataset formed by images of 100 $\times$ 100 resolution in which its training set has approximately 53000 images divided in 103 classes. These images were obtained by placing a white sheet as background and having large illumination differences. Supermarket Produce [33] is a dataset that also contains images of fruits, however, with only 11 categories and approximately 2000 images of 1024 $\times$ 768 pixels. Supermarket Produce also has variation in lighting and poses in which the amount of elements vary in the composition of the image. Additionally, this dataset contains images with packed fruits, which causes reflection of luminosity. Due to the variation in the number of classes between these two datasets, we have selected 9 common labels and, consequently, reducing the amount of images. In a specific configuration, Fruits-360 has several classes containing specific types of red apples and, in our experiments, we considered as one single class.

b) ***Object****:* Amazon and Webcam [34] are part of the same dataset, widely used in domain adaptation tasks. Amazon is formed by images downloaded from the web that have white background and studio lighting, totaling 2817 images of 300 $\times$ 300 pixels categorized into 31 categories. Webcam has exactly the same categories, however, the images (795 in total) have resolution variation from one example to another, presence of noises, and artifacts in the background with great difference of illumination. The Fig. 5 presented some examples from these two datasets.

c) ***Skin lesions****:* HAM10000 [35] is a dataset that contains skin lesions images of 600 $\times$ 450 pixels of resolution with 7 distinct classes. With only two classes, PH2 [36] has only 200 images of 768 $\times$ 574 pixels. Both datasets have variance of brightness and confusing objects, such as hair, bubbles, and black margins, in which the malignancy of a lesion is defined by the uniformity of shapes, colors, and texture, as shown in Fig. 6. We consider only the two common categories of the two datasets (nevus and melanomas), reducing the amount of HAM10000 images to approximately 7800 images.

d) ***Photos****:* Corel1000 [37] is a dataset widely used in classification tasks because it has fully balanced classes. Comprising 10 classes of 100 examples, its images have a resolution of 384 $\times$ 256 pixels. In the experiments performed with Corel1000 we splitted randomly the images in training and test sets, in the proportion of 80/20, respectively. Some images that composed this dataset are displayed in the Fig. 7.

With these datasets, Fruits-360, Amazon, and HAM10000 were used as source for Supermarket Produce, Webcam, and PH2, respectively. Due to the input configured in ResNet-50, all images were resized to 224 $\times$ 224 pixels.

B. Fine-tuning setup

For a more effective comparison of the representativeness capability from ResNet-50 pre-trained with ImageNet [8], we also fine-tuned the same network with one dataset source. The fine-tuning setup was basically the same used in the original ResNet-50 training: SGD with mini-batch size of 256, learning rate of 0.1 with weight decay of 0.0001, and momentum of 0.9 [19] during 100 epochs. However, only the last seven layers were allowed to adapt with the new domain. This configuration was adopted with the objective of maintaining the descriptors from initial layers frozen, offering a better observation of the global and fusion performances.

C. Evaluation

With the feature spaces provided by ResNet-50 (pre-trained with ImageNet or fine-tuned with the source domain), TCA was applied aiming to highlight the similarities and reduce the discrepancies between the data distributions. Five different scenarios were tested, choosing 256, 192, 128, 96, and 64 attributes with RBF kernel. After the feature transformation by TCA, we employed Linear SVM to verify how linearly separable are the classes in the feature space [38]. Linear SVM was chose due to capacity of guarantee stronger learning and for ensuring a more restricted bias [39]. To train the classifier, only the feature space obtained from the source dataset is applied, without any knowledge of the test examples.

In summary, we have features extracted from the same layer for both (source and target) datasets. One dataset is used to train the classifier (source) and the other one to test it (target). The accuracy of classfication is used to evaluate and compare the methods. The same protocol is used when evaluating global features; local features; and combining those features.

V. Results and Discussion

Based on the chosen datasets, fine-tuning setup, and evaluation, Tables I-IV present the results obtained for our novel. We emphasize here that the objective of this experiment is not to overcome the competing methods in the literature, but rather to present the importance of more investigation of initial layers from a CNN that frequent is neglected in the conception of the methods. Also, due to the application of TCA the fusion and global performances have the same number of attributes.

Table 1 shows that, on average, the global descriptor is better when the network does not incorporate the new semantics contained in Fruits-360 (30.79% vs. 27.17%). However, in general terms, either with fine-tuning or without, the multi-layer fusion performance is highly applicable to this dataset. Considering the average of multi-layer fusion features selection (PCA, Flatten Pooling, and Pooling 2D), when the

Fig. 4. Examples from: (top) Fruits-360; (bottom) Supermarket Produce. Although both datasets contain only fruits, they differ in the amount of elements in each image, the size of the objetcs, and illumination. From the left to the right: red apple; green apple; kiwi; lime; nectarine; orange; peach; pear; and plum.

Fig. 5. Examples from: (top) Amazon; (bottom) Webcam. Although both datasets contain office items, they differ in 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.

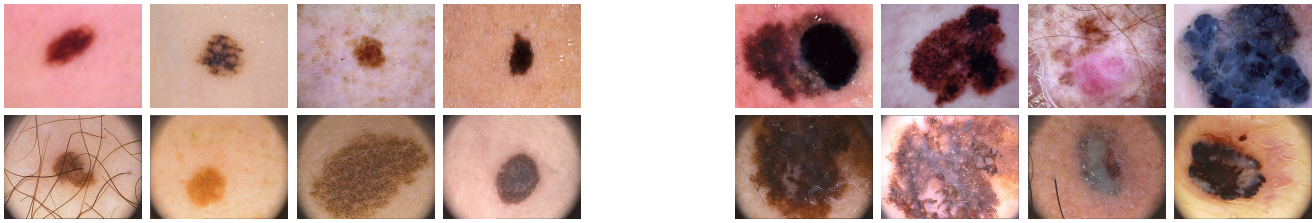

Fig. 6. Examples from: (top) HAM10000; (bottom) PH2. Although both datasets contain skin lesions, they differ due to different margins and presence of clutter as hair. The fisrt four images (left) indicates common nevus and the others (right) represent melanomas.

Fig. 7. Examples from Corel1000 dataset. From the left to the right: food; indian; beach; architecture; bus; dinosaur; elephant; flower; horse; and montain.

TABLE I

CLASSIFICATION ACCURACY (%) OF SUPERMARKET PRODUCE DATASET COMPARING FEATURE EXTRACTION FROM RESNET50 PRE-TRAINED WITH IMAGENET VERSUS RESNET-50 FINE-TUNED WITH FRUITS-360 DATASET. VALUES IN BOLD (FUSION) REPRESENT HIGHER ACCURACY WHEN COMPARED WITH GLOBAL RESULTS. THE * INDICATES WHEN THE FINE-TUNING PERFORMANCE OVERCOMES ITS RESPECTIVE IMAGENET RESULT.

Supermarket Produce	**Features**	**Global**	**Fusion 1th block**			**Fusion 2th block**			**Fusion 3th block**		
			PCA	**Flatten**	**Pool. 2D**	**PCA**	**Flatten**	**Pool. 2D**	**PCA**	**Flatten**	**Pool. 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***

TABLE II

CLASSIFICATION ACCURACY (%) OF WEBCAM DATASET COMPARING FEATURE EXTRACTION FROM RESNET50 PRE-TRAINED WITH IMAGENET VERSUS RESNET-50 FINE-TUNED WITH AMAZON DATASET. VALUES IN BOLD (FUSION) REPRESENT HIGHER ACCURACY WHEN COMPARED WITH GLOBAL RESULTS. THE * INDICATES WHEN THE FINE-TUNING PERFORMANCE OVERCOMES ITS RESPECTIVE IMAGENET RESULT.

Webcam	**Features**	**Global**	**Fusion 1th block**			**Fusion 2th block**			**Fusion 3th block**		
			PCA	**Flatten**	**Pool. 2D**	**PCA**	**Flatten**	**Pool. 2D**	**PCA**	**Flatten**	**Pool. 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 III

CLASSIFICATION ACCURACY (%) OF PH2 DATASET COMPARING FEATURE EXTRACTION FROM RESNET50 PRE-TRAINED WITH IMAGENET VERSUS RESNET-50 FINE-TUNED WITH HAM10000 DATASET. VALUES IN BOLD (FUSION) REPRESENT HIGHER ACCURACY WHEN COMPARED WITH GLOBAL RESULTS. THE * INDICATES WHEN THE FINE-TUNING PERFORMANCE OVERCOMES ITS RESPECTIVE IMAGENET RESULT.

PH2	**Features**	**Global**	**Fusion 1th block**			**Fusion 2th block**			**Fusion 3th block**		
			PCA	**Flatten**	**Pool. 2D**	**PCA**	**Flatten**	**Pool. 2D**	**PCA**	**Flatten**	**Pool. 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

TABLE IV

CLASSIFICATION ACCURACY (%) OF COREL1000 DATASET (TEST SET) COMPARING FEATURE EXTRACTION FROM RESNET50 PRE-TRAINED WITH IMAGENET VERSUS RESNET-50 FINE-TUNED WITH COREL1000 DATASET (TRAINING SET). VALUES IN BOLD (FUSION) REPRESENT HIGHER ACCURACY WHEN COMPARED WITH GLOBAL RESULTS. THE * INDICATES WHEN THE FINE-TUNING PERFORMANCE OVERCOMES ITS RESPECTIVE IMAGENET RESULT. THE DATASET WAS SPLITED IN 80% FOR TRAINING AND 20% FOR TEST.

Corel1000	**Features**	**Global**	**Fusion 1th block**			**Fusion 2th block**			**Fusion 3th block**		
			PCA	**Flatten**	**Pool. 2D**	**PCA**	**Flatten**	**Pool. 2D**	**PCA**	**Flatten**	**Pool. 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*

network is fine-tuned the performance increases in 8.36% with the first residual block. Specifically, these results show that Fruits-360 and Supermarket Produce are datasets with predominantly low-level features, such as shapes and edges. This is evidenced when the global descriptor has its performance reduced with fine-tuning. Analyzing the methods of multi-layer fusion features selection it is observable that Pooling 2D is practically constant in all layers, 29% without fine-tuning and 38% with fine-tuning. PCA has a larger variation when the network is fine-tuned (20.66%), reaching its peak in the first block (41.46%), then gradually decays. However, Flatten Pooling has a better performance in the second block,

as illustrated in the Fig. 8.

Considering Amazon (source) and Webcam (target) in Table II, we noticed a decrease in the performance of the multi-layer fusion features, although it is better at about 1.25% on average when compared with global (52.52% vs. 51.25% with ImageNet and 51.58% vs. 50.36% with fine-tuning). This is because Webcam is a dataset with greater variance (background, perspective, and presence of clutter), requiring more representation from the global descriptor. However, the multi-layer fusion features still offers accuracy gain on average using Flatten Pooling (55.75% vs. 51.25%) in the first block without fine-tuning. Similar to the results of Supermarket Produce, Pooling 2D remains practically constant and PCA has better performance with fine-tuning in the second block.

Images of skin lesions present different texture, being this a decisive factor to diagnose an injury as malignant or not. Consequently, and evidenced by the results presented in Table III, the adoption of global and local features do not increase the accuracy (85.7% vs. 85.62% for ImageNet and 86.0% vs. 85.67% applying fine-tuning network) on average. Despite a slight superiority in few multi-layer fusion features methods (PCA, Flatten Pooling, and Pooling 2D), all of them presented themselves in an equivalent form in all residual blocks.

Contrary to previous results of Webcam and PH2, PCA stands out in relation to the global descriptor and other multi-layer fusion features methods for Corel1000, in Table IV. This is because all examples, both training and testing, have similar data distribution due to they belong to the same dataset. Consequently, the components selected in the training set are practically the same as those that should be selected in the test set, increasing the accuracy. However, almost all fusion methods excel at their global performance, except Pooling 2D in the second and third block with fine-tuning, which maintains the same result (94.4%).

To show the importance of the most initial layers of a CNN for representativeness of the feature spaces, Fig. 9 presents a comparative of differences between the performance of the local descriptors in relation to the global one. These values are the average among TCA performances (256, 192, 128, 96, and 64 attributes). It is interesting to note that the first block offers better performance than the others and, as the layers become more mid-level, the accuracy decreases gradually. These results confirms that layers at the end of the network offer more representative feature spaces when they are used alone. Evidently, the performance gain by performing fusion of multiple layers activations is achieved with the increase of computational cost. PCA presents greater computational complexity due to the need to evaluate the variance in all features present in the space. However, the other two methods are simpler, requiring only delimitation of a region and calculation of the average among these values. Consequently, its computational cost is more linked to the number of examples in the training and testing set than to the complexity of the task. The adoption or not of this novel is directly related to the need for precision in the classification, in cases where the correctness is more important than the computational cost.

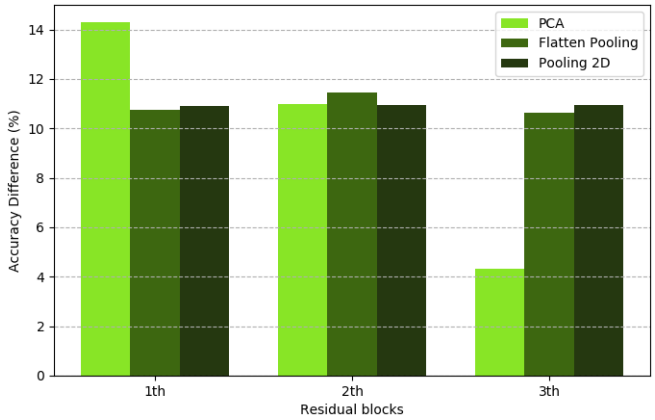

Fig. 8. Classification accuracy (%) difference on average between multi-layer fusion and global results using Fruits-360 as source dataset and Supermarket Produce as target dataset in ResNet-50 fine-tuned.

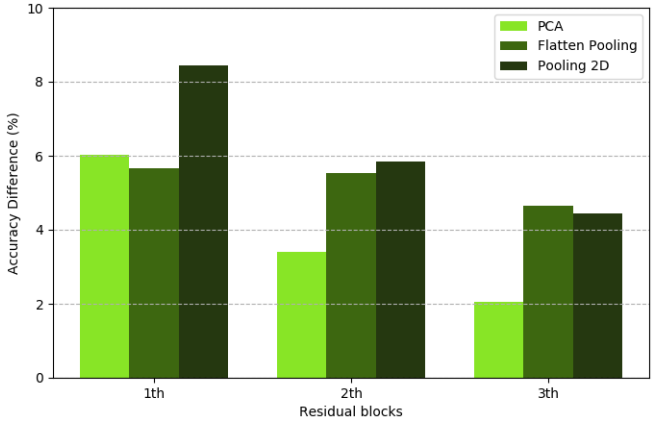

Fig. 9. Classification accuracy (%) difference on average between local and global results using Fruits-360 as source dataset and Supermarket Produce as target dataset in ResNet-50.

VI. Conclusion

In this study we investigate descriptors from low-level layers of a convolutional neural network to complement those of top layers in scenarios of transfer learning. Performing a fusion and alignment of data distributions from local (first layers activations) and global (top layers activations), our novel approach has been evaluated in different datasets with different domains. We have shown that images with well behaved objects are better classified by merging attributes from different layers. For images with more clutter or with larger intra-class variance, the global CNN descriptors are more adequate and the addition of local information becomes less effective. We show that even with few examples used to fine-tune the network, our approach significantly improves transfer learning when compared to using ImageNet initialized weights. Our results represent a step towards improving a more efficient feature extraction, taking into account local and global CNN descriptors. We also offer guidelines for future studies to investigate initial and mid-level layers. A methodology to identify most discriminative layers is also a matter of future

investigation. Future work can also study complementary information from handcrafted features in comparison with global and local CNN features.

VII. Acknowledgments

This study was financed by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, FAPESP (grant #2018/22482-0), CNPq (grant 307973/2017-4), and CEPID-CeMEAI (FAPESP grant #2013/07375-0).

References

[1] 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.

[2] M. M. Ghazi, B. Yanikoglu, and E. Aptoula, "Plant identification using deep neural networks via optimization of transfer learning parameters," *Neurocomputing*, vol. 235, pp. 228–235, 2017.

[3] K. Nogueira, O. A. Penatti, and J. A. dos Santos, "Towards better exploiting convolutional neural networks for remote sensing scene classification," *Pattern Recognition*, vol. 61, pp. 539–556, 2017.

[4] M. d. F. O. Baffa and L. G. Lattari, "Convolutional neural networks for static and dynamic breast infrared imaging classification," in *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE, 2018, pp. 174–181.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] H. Ravishankar, P. Sudhakar, R. Venkataramani, S. Thiruvenkadam, P. Annangi, N. Babu, and V. Vaidya, "Understanding the mechanisms of deep transfer learning for medical images," in *Deep Learning and Data Labeling for Medical Applications*. Springer, 2016, pp. 188–196.

[10] 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.

[11] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear cnn models for fine-grained visual recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1449–1457.

[12] S. Sadigh and P. Sen, "Improving the resolution of cnn feature maps efficiently with multisampling," *arXiv preprint arXiv:1805.10766*, 2018.

[13] Y. Pan, T. Yao, H. Li, and T. Mei, "Video captioning with transferred semantic attributes," in *CVPR*, vol. 2, 2017, p. 3.

[14] 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.

[15] Z. Shi, H. Hao, M. Zhao, Y. Feng, L. He, Y. Wang, and K. Suzuki, "A deep cnn based transfer learning method for false positive reduction," *Multimedia Tools and Applications*, pp. 1–17, 2018.

[16] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, p. 115, 2017.

[17] 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.

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

[19] 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.

[20] W. M. Kouw and M. Loog, "A review of single-source unsupervised domain adaptation," *arXiv preprint arXiv:1901.05335*, 2019.

[21] 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.

[22] Y. Zheng, J. Huang, T. Chen, Y. Ou, and W. Zhou, "Cnn classification based on global and local features," in *Real-Time Image Processing and Deep Learning 2019*, vol. 10996. International Society for Optics and Photonics, 2019, p. 109960G.

[23] Q. Yu, X. Chang, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "The devil is in the middle: Exploiting mid-level representations for cross-domain instance matching," *arXiv preprint arXiv:1711.08106*, 2017.

[24] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 447–456.

[25] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1395–1403.

[26] Y. Chen, S. Duffner, A. Stoian, J.-Y. Dufour, and A. Baskurt, "Pedestrian attribute recognition with part-based cnn and combined feature representations," in *VISAPP2018*, 2018.

[27] Z. Ge, S. Demyanov, B. Bozorgtabar, M. Abedini, R. Chakravorty, A. Bowling, and R. Garnavi, "Exploiting local and generic features for accurate skin lesions classification using clinical and dermoscopy imaging," in *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*. IEEE, 2017, pp. 986–990.

[28] J. Mišeikis, I. Brijacak, S. Yahyanejad, K. Glette, O. J. Elle, and J. Torresen, "Transfer learning for unseen robot detection and joint estimation on a multi-objective convolutional neural network," in *2018 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*. IEEE, 2018, pp. 337–342.

[29] M. M. Kalayeh, E. Basaran, M. Gökmen, M. E. Kamasak, and M. Shah, "Human semantic parsing for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1062–1071.

[30] J. Wen, R. Liu, N. Zheng, Q. Zheng, Z. Gong, and J. Yuan, "Exploiting local feature patterns for unsupervised domain adaptation," *arXiv preprint arXiv:1811.05042*, 2018.

[31] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5542–5551.

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

[33] 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.

[34] 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.

[35] 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.

[36] 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 *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*. IEEE, 2013, pp. 5437–5440.

[37] J. Z. Wang, J. Li, and G. Wiederhold, "Simplicity: Semantics-sensitive integrated matching for picture libraries," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 9, pp. 947–963, 2001.

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

[39] V. N. Vapnik, "An overview of statistical learning theory," *IEEE transactions on neural networks*, vol. 10, no. 5, pp. 988–999, 1999.