Data Augmentation Guidelines for Cross-Dataset Transfer Learning and Pseudo Labeling

Fernando Pereira dos Santos, Gabriela Salvador Thumé, Moacir Antonelli Ponti
ICMC — Universidade de São Paulo (USP), São Carlos, SP, Brazil
Email: {fernando_persan,gabithume}@alumni.usp.br,ponti@usp.br

***Abstract*—Convolutional Neural Networks require large amounts of labeled data in order to be trained. To improve such performances, a practical approach widely used is to augment the training set data, generating compatible data. Standard data augmentation for images includes conventional techniques, such as rotation, shift, and flip. In this paper, we go beyond such methods by studying alternative augmentation procedures for cross-dataset scenarios, in which a source dataset is used for training and a target dataset is used for testing. Through an extensive analysis considering different paradigms, saturation, and combination procedures, we provide guidelines for using augmentation methods in favor of transfer learning scenarios. As a novel approach for self-supervised learning, we also propose data augmentation techniques as pseudo labels during training. Our techniques demonstrate themselves as robust alternatives for different domains of transfer learning, including benefiting scenarios for self-supervised learning.**

I. Introduction

Deep Networks were able to thrive in computer vision in the last decade due to the availability of computational power and large training datasets [1]. With those requirements sufficiently met, Convolutional Neural Networks (CNNs) are widely applied for image classification tasks, initially classification of photographic images [2] and later extended to different domains. These networks are characterized by having a hierarchical architecture, in which the output of a given layer serves as an input for a posterior one. Also, each layer has specific functionalities, which the two most relevant are convolution, filtering the input, and pooling to reduce the data dimensionality [3]. Due to this hierarchical structure, one of the great advantages of CNNs is their capacity to represent different levels of abstraction. The first layers provide low-level features (shapes, edges, and colors) while the last layers are associated with high-level features, mostly semantics [4]. Another advantage of CNN filters is their capacity for generalization. By training a CNN using a sufficiently large and diverse training data, such as ImageNet [2], those filters can be useful for other datasets, favoring transfer learning [5]. For photographic images, the better the model performs using ImageNet, the better it tends to transfer to other datasets [6].

In the context of transfer learning [7], [8], two approaches are more common: extracting features directly from a pre-trained CNN to an external classifier and fine-tuning a pre-trained CNN [3]. In the first approach, one can choose which layer offers the best descriptor for some data distribution and

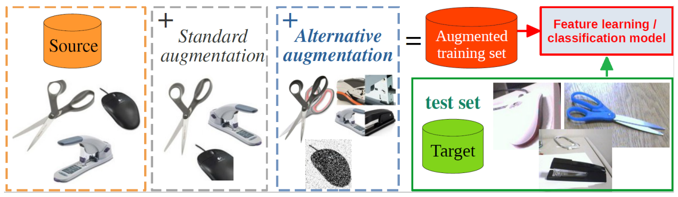

Fig. 1. Considering a source dataset (as training set), we apply the standard augmentation or our alternative methods to increase the diversity and representativeness of the final training set. This set is used to fine-tune the classification model, obtained from a pre-trained model, followed or not by self-supervised learning. Thus, a target set (another dataset from the same domain of training set) is evaluated on the classification model.

perform feature extraction [9]. However, when performing fine-tuning, the last layer (the prediction one) is redefined, then the remaining parameters are initialized with the pre-trained weights, and adaptation is resumed for a number of epochs using a training set for the new task. During fine-tuning itself, characteristics of the new domain are incorporated, promoting adaptability of the network weights to the new context and increasing learning. The success of network fine-tuning depends on sample representativeness, depth of architecture network, and complexity of the task [10], [11]. Since recent architectures are often deep, and task difficulty is not easily measurable [10], increasing the size of the training set becomes a manner to ensure sample representativeness. Thus, data augmentation may be used in this context to create new examples by processing original images [12], which are incorporated in the training set, increasing the diversity [13].

Data augmentation techniques were shown to be effective in several applications [13]–[18]. Even for large datasets, such as ImageNet [2], augmenting the training set is a common practice that showed to improve classification results [19], [20]. Image augmentation has been extensively studied for scenarios within the same domain with respect to training and testing sets [16], [17], while the literature still lacks evidence for cross-dataset scenarios (using one dataset to train or fine-tune the network and another to test it), where different forms of acquisition, such as perspective and number of objects present additional challenges. Also, methods of image manipulation are mostly restricted to those available in software packages, which perform subtle processing in individual images. Another attempt to better adapt a pre-trained CNN to a new domain or task is through self-supervised learning. In this context, new synthetic images are generated following some heuristic and pseudo-labels are distributed according to this methodology [21]. For instance, given an initial training set of images, one can rotate each image in different orientations, each one representing a different class [22]. Then, after initial training to learn the concepts inherent of the new domain, the network is fine-tuned with the original data and respective original labels. Hence, the performance may be increased significantly without the required cost for labeling new examples. Also, generalization in terms of external validation is an important open question. In fact, usually, the augmentated data used in training comes from the same source dataset/domain used in test stage, while cross-dataset scenarios are more realistic and challenging [11], [23].

Aiming to investigate the behavior of data augmentation techniques for scenarios with different datasets, our general proposal is depicted in Figure 1. We demonstrate that by adding more diversity to the training set via alternative augmentation methods, the model learns features that transfer better to other datasets. In order to demonstrate that, we used a cross-dataset approach: a source set (and augmentation) is used to fine-tune the pre-trained network and another one (target set) for testing only. In addition to the standard procedures, our alternative augmentation techniques (see Section III-B) improves the model performance. Beyond the conventional data augmentation approach for the purpose of improving performance, we thoroughly analyze the saturation of training set size generated from those techniques. As a manner to enhance the network fine-tuning step, we also investigated the influence of self-supervision as a previous step. In this complementary approach, each augmented set receives a pseudo-label.

Therefore, our contribution includes: (i) six alternative techniques for data augmentation beyond standard ones, including methods that simply apply some image transformation and methods that perform combinations among images from the same class; (ii) a study with self-supervised learning using pseudo-labeling, in which different data augmentation techniques act as distinct labels, providing a significant improvement to network performance; and (iii) extensive cross-dataset evaluation of standard techniques and our alternative methods, increasing the training set size, verifying the saturation, and sensibility to self-supervised learning.

II. Data Augmentation and Self-supervised Learning Context

Data augmentation is a popular strategy to improve classification results, including on large annotated datasets [2] and when using state-of-the-art algorithms [17], providing increased diversity through a single dataset or merging distinct sets that share the same labels to generate new instances [13]. An extensive survey by Shorten and Khoshgoftaar [12] defined a taxonomy for data augmentation techniques performed on Deep Learning models, which has three main categories: image manipulation; deep learning approaches; and meta-learning. Potential benefits were reported: an increase in the amount of data available and data variability; convergence improvement; and reduced overfitting. However, the survey could not find a consensus about the best strategy for combining techniques and did not report papers investigating cross-dataset and transfer learning scenarios.

Image manipulation for data augmentation consists of generating modified versions of images in some source dataset. Such transformation techniques are chosen considering the dataset context and often ensuring the creation of plausible versions, avoiding transformations that may alter the actual label of the image. For instance, by applying a 180-degree rotation operation as augmentation for handwritten digits, digits six and nine can have their label swapped [24]. In the literature, image manipulation, such as rotation and shift (more details in Section III), are considered standard methods due to their low computational cost and easy implementation, with availability in most software packages [12]. Such standard methods perform subtle changes in pixel content so that not to change the image in a manner its label would be altered [12]. Alternatives in this sense include simulating occlusion [18], in which some pixels are erased within a rectangle region and receive random values, and a crop-and-patch method [25], which mixed images and provided results that are similar to more complex methods that rely on learning.

Data augmentation methods can also be conceptually grounded by learning approaches [24]. Often Generative Adversarial Networks (GANs) are applied to generate instances based on the initial set through adversarial learning [26]. However, these networks are difficult to train, involving high computational cost. Considering meta-learning, Wang and Perez [17] proposed the Neural Augmentation, a method that allows a neural network to learn which data augmentation method is better to reduce classification loss for a given dataset. This architecture is composed of two networks, one performs the classification properly while the other performs the combination of two images of the same class, resulting in an image of six channels. During training, the loss functions are combined to improve both branches simultaneously. In their results, they indicated that simple transformations, such as cropping, rotating, and flipping, are very effective when used alone, consuming lower computing time. Another meta-learning data augmentation, a Reinforcement Learning algorithm called AutoAugment, was proposed by Cubuk et al. [16], which automatically searches the best policy for the neural network to yields the highest accuracy. Sixteen operations were performed, among them are: equalize; rotate; brightness; posterize; translate; and others. In both studies [16], [17], image manipulation-based augmentation combined with learning-based augmentation was shown to increase accuracy and reduced overfitting.

Another concept to reduce possible overfitting during the training process is self-supervision. Self-supervised learning presents itself as an unsupervised technique for supervised networks [22], due to the creation of pseudo-labels. Thus, synthetic data are generated to increase the representativeness of the training set following a methodology that defines those labels. In this context, different self-supervised methodologies can be applied in an attempt to improve the generalization and performance of a deep network. Concerning images, some alternatives that are presented in the literature include: rotation [27] to recognize different orientations; exemplar [28], where each image represents a class, however, several generating image methods must be employed; and jigsaw [29], which aims to teach the spatial localization of patches. However, these techniques essentially have a specific learning objective, such as orientation or location.

Based on these related studies, recent literature indicates image manipulation still plays a significant role as data augmentation, with low computational complexity, being complementary to or compared with learning techniques which, generally, have high computational costs and require themselves sufficiently large training sets. Therefore, aiming to maintain low complexity in the generation of images and well performance behaviors, we investigated six alternative methods. These methods are compared, in our results section, with the standard methods in the literature for cross-dataset scenarios, considering the saturation of training set size. In addition to the saturation analysis, we also propose a novel methodology for self-supervised learning. Taking advantage of the new images, generated by data augmentation techniques investigated here, our approach embraces several methods in a single framework. Unlike existing approaches, we consider that each image set generated by a single data augmentation technique becomes a training class. Therefore, exploring several methods simultaneously does not restrict the learning to a unique scenario.

III. Methodology

Our pipeline is composed by the choice of a data augmentation method and the number of synthetic images that will be generated from a single instance of the source training dataset. Considering a pre-trained network (described in Subsection IV-C), we fine-tune the network with this resulting set and validate its predictive capability with the target dataset, as described in the Algorithm 1. In addition, as a previous step of fine-tuning, we may or may not apply self-supervision (detailed in Subsection IV-D).

We describe in the following the standard augmentation methods, as well as the alternative ones we propose to complement and improve cross-dataset feature learning. In all cases, we respected the characteristics of the datasets used in the experiments, by setting parameters so that the image manipulation does not alter the resulting image's label [30]. Six alternative image manipulation methods are described. Our proposed data augmentation techniques are divided into two different categories: methods that use a single source image as input (Blur, Correlated Noise, and Sharpening); and methods that combine a pair of images from the same label (Blend, Threshold, and Visual SMOTE). Examples of such methods are shown in Figure 2.

Algorithm 1 - Methodology procedure

```
    $T_r$: Training set;
    $T_e$: Testing set;
    $D$: Data Augmentation techniques;
    $n$: Number of images to be generated for each image in $T_r$;
    $i$: Number of methods of $D$ to be selected;
    $f$: Pre-trained CNN model.

1: Generate $n$ images ($A_i$) using $D$ for each image in $T_r$
2: if self-supervised learning then
3:     All images in $T_r$ receive a pseudo-label (zero)
4:     Each images in $A_i$ receive the pseudo-label $i$, $i > 0$
5:     Self-supervised training of $f$ using $T_r$ and $A_i$
6: end if
7: Fine-tune $f$ using all images from $T_r$ and $A$ (original labels)
8: Evaluate $f$ using all images of testing set $T_e$
```

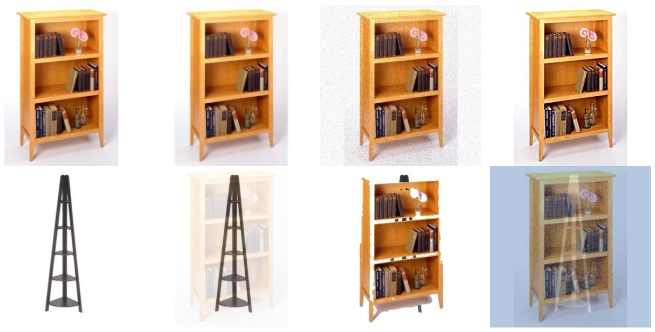
Fig. 2. Examples of outputs from: (top) original one; Blur; Correlated Noise; Sharpening; and (bottom) pair from the same class; Blend; Threshold; Visual SMOTE.

A. Standard Methods

i) Flip: Flipping images to augment training data is one method to improve performance by oversampling. This technique simulates different points of view of an image. We performed random flips on both horizontal and vertical axes.

ii) Rotation: One of the most widely used operations and considered to be an oversampling technique [12]. Original images were rotated, generating new ones with different orientations. We performed random rotation from 0-degree to a 90-degree clockwise range in our experiments.

iii) Shift: Shift augmentation artificially creates horizontal and vertical shifted versions of the training data. Applying a shift to an image means moving all pixels horizontally or vertically, which we performed of a maximum of twenty percent for the width and height of the image.

B. Alternative Methods

i) Blur: Image blurring, also known as image smoothing, is useful to filter an image, intentionally removing noise and other not relevant details. Common methods can blur the edges, which is an unwanted behavior when we want to generate new images, as some relevant information can be removed. On the other hand, an operation called *bilateral filter* can be used to preserve sharp edges but at the same time blur the image. It replaces the pixel value with the average of neighboring pixels of similar intensity [31]. It is a weighted average of the intensities while considers the difference between the values of neighbors (nearby pixels) to preserve the edges. Thus, for one pixel to influence another, it must be close in the coordinate space and have similar intensity.

ii) Correlated Noise: Correlated noise occurs in the photon count of optical devices and follows the Poisson distribution, which represents the number of occurrences of an event at any given time. A common implementation is based on Poisson-distributed random numbers, as developed by Knuth [32], and adapted to a matrix of pixels. To calculate the noisy value in a pixel, the pixel is considered the average of this distribution. Thus, there will be barely any noise for dark intensities. On the other hand, at intensities close to 255, the resulting intensity will be higher. For adding this noise to an image, no parameters are provided due to noise being calculated for each pixel.

iii) Sharpening: A well-known image sharpening technique that seeks to emphasize intensity transitions called *unsharp masking* was used. It starts by creating a blurry image from the original, then it subtracts the blurred one from the original, and adds the image result to the original one, given a weight of k. Gaussian blur was the method used to produce the blurred image, and we applied the weight of $k = 1$ to highly emphasize the image.

iv) Blend: This method generates a new image by calculating the weighted sum of two images. From a pair of images, we draw an α value below 100%; then, we calculate β, which is equivalent to the remaining value to complete 100% (100% - α); and lastly, each pixel of the new image is the result of multiplying the pixel of the first image by β and adding it to the pixel of the second image times α. The parameters α and β are chosen at random. A value between 10% and 90% is chosen for α so any image contributes with at least 10% for the final result and β is calculated according to its value.

v) Threshold: The resulting image is a composition of the foreground (scene object) of an image and the background of another image. We use the OTSU threshold method to find the binary foreground and then we apply morphological operations to improve the resulting mask, which is subsequently used to apply the colored foreground to the image used as background.

vi) Visual SMOTE: SMOTE (Synthetic Minority Oversampling Technique) is a dataset balancing method often applied in images after feature extraction [33]. It creates a new sample by performing a combination of instances that are close in the feature space. A visual alternative was adapted to pixel-level and performed between two images of the same label. The difference between the pixels of the two images is calculated; multiplied by a random number in the range of $[0 - 1]$; and added to the original image. We note the effects are similar to the Blend method, but the random component at each pixel creates more diverse images in terms of color combination and background.

IV. Experiments Setup

A. Datasets

We evaluated two domains, representing the classification task of objects and fruits. A pair of distinct datasets are used for each domain. The largest dataset is used as a source to fine-tune the CNN, where we applied the data augmentation methods and self-supervised learning, while the smaller one serves as a target only for testing. The diversity of domains with different styles, scene composition, and degrees of difficulty enriches the contribution of this study for data augmentation and self-supervised learning. All datasets present somewhat unbalanced classes, but none is severely imbalanced. Four datasets were used in our experiments:

a) Objects: Amazon and Webcam [34] are datasets with office objects containing the same 31 categories. Amazon consists of e-commerce images of products with controlled illumination and constant background, while Webcam was captured with illumination variations, clutter (including background and confounding objects), and noise. Amazon has 2817 images of 300 $\times$ 300 pixels (used as the training set) and Webcam has 795 examples with varying resolution (used as the test set).

b) Fruits: Supermarket Produce [35] is a dataset with 15 classes and around 2600 images of 1024 $\times$ 768 pixels, containing a variation of lighting, poses, number of objects in the image, and including images with packed fruit. Within the same domain, FIDS30 [36] has 31 classes and 972 images with different resolutions, also having variations of lighting and number of objects in the same photo. In our experiments, we considered a subset of both datasets, containing only seven common classes between them: kiwis; limes; oranges; peaches; pears; plums; and watermelons. Hence, we selected 1206 images of Supermarket Produce (training set) and 221 images from FIDS30 (test set).

Due to the input layer of CNN used in our experiments, all images were resized to 224 $\times$ 224 pixels, as shown in Fig. 3. Considering the training sets (Amazon and Supermarket Produce), we generated new synthetic images using the data augmentation techniques and all original set.

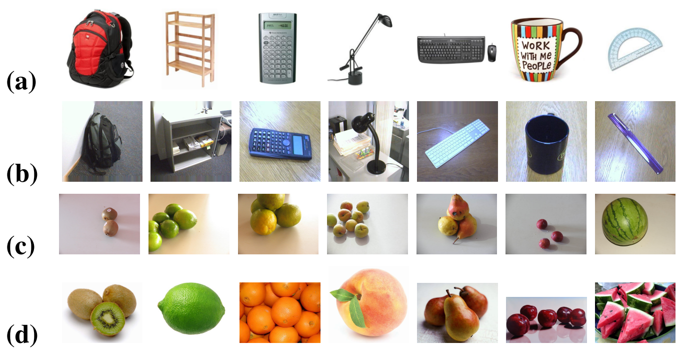

Fig. 3. Original examples from: (a) Amazon; (b) Webcam; (c) Supermarket Produce; and (d) FIDS30. For objects domain, from left to right: backpack; bookcase; calculator; desk lamp; keyboard; mug; and ruler. For fruits domain, from left to right: kiwi; lime; orange; peach; pear; plum; and watermelon.

B. Synthetic sets

Considering a data augmentation technique D and a training set T_r for the target task, **each image belonging to T_r will provide n resulting image**s. Therefore, if T_r has k images, the new synthetic set A will have $A = k * n$ examples. Hence, we may generate different synthetic set sizes, given a single method at a time and the number of variations. During the CNN fine-tuning, the final training set size will be $F = T_r + A$. In our approach, n ranges from 0 to 5. When $n = 0$, only the original set is used as a training set. This scenario is our baseline. When $n = 1$, the training set is formed by the original set and by one additional image from each original example, i.e the final set has twice as many examples of the baseline. This procedure is scalable to the other values of n. It is important to note that each synthetic set was generated using only a single technique and by the full source set. Accordingly, several synthetic sets were generated.

C. CNN Backbone

Residual Networks [20] introduced the concept of residual blocks with "skipping layers" to allow training for deeper networks. After the last residual block, an Average Pooling is applied, which is followed by a dense output layer. Its 50-layer version, ResNet50, is widely used for transfer learning and remains among the state-of-the-art for many applications, including domain adaptation [37], [38], and often used to evaluate data augmentation [16], [18]. We fine-tuned the network changing only the last layer during 1000 epochs with SGD optimizer (learning rate of 0.01 and momentum of 0.0001), using the Cross-entropy loss function [39], [40], and a 32 batch size (stipulated optimal value [41]), allowing all layers to update. The test performance is measured using accuracy and only the target set.

D. Technical Self-supervised

We apply self-supervision as a manner to leverage the network fine-tuning step. Our approach for self-supervised learning is to consider each set, generated by a data augmentation technique, as a distinct label. Initially, a mandatory set is the initial training set, i.e the original images without their respective labels (Amazon or Supermarket Produce), which all images assume an unique label (zero). To complement these images to form the training set, one can choose as many data augmentation techniques as desired, each one will assume a new unique label (label one until n, each one for a different technique). Consequently, if n methods are selected, the number of self-labels will be $n+1$ (due to the original set). Therefore, considering c the number of self-labels belonging to the final training set, $c = n + 1$, where $n > 0$. Self-supervised training is carried out following the same specifications as fine-tuning, SGD optimizer with a learning rate of 0.01 and momentum of 0.001, Cross-entropy loss function, and 32 batch size during 1000 epochs. After, fine-tuning is performed using the same training set, changing the pseudo-labels for the original ones.

V. Results and discussion

Our results and discussions are reported in three parts: (i) data augmentation techniques performance, where we analyze the accuracy of each technique; (ii) data augmentation for self-supervised learning, where we analyze the behavior of data augmentation techniques in relation to the sensitivity of self-supervision; and (iii) augmented and self-supervised computational costs.

A. Data Augmentation techniques performances

For each data augmentation technique, both the alternative and standard methods, we ranged the number of examples generated from each image from 1 to 5. Contextually, when $n = 1$ each image in the original set provided 1 additional image, and so on, successively.

Considering the objects domain (Amazon as training set and Webcam as test set), the accuracy achieved using only the original training set, without data augmentation ($n = 0$), was 40.62%. Based on the performance achieved applying data augmentation methods (see Table I), we can see that the best accuracy was obtained using the Correlated Noise (68.93% for $n = 2$) and the worst accuracy was Shift (51.44% when $n = 1$). Consequently, the performance gain due to data augmentation techniques ranged from 10.82% to 28.31%. As expected, performances for $n = 1$ are not as satisfactory as for other values of n, despite the expressive performance gain (about 58.23% on average). In addition, standard methods provide inferior results to the proposed methods. The best performance of these methods is achieved with Shift (66.41% with $n = 4$). Of the proposed methods, only Blend (-1.89% lower when $n = 2$) and Visual SMOTE (0.13% when $n = 3$) do not provide better accuracy. In the saturation view, only Flip reaches its better performance with $n = 5$. All other methods suffer a training set saturation when the same image provides many equivalent examples. This degradation is more sensitive in methods that combine images, such as Blend, Threshold, and Visual SMOTE. Specifically for this domain, we have classes that have about 30 examples, providing the combinations (randomly) less distinct from each other. Thus, the variability does not become a relevant factor for high n in these techniques. Despite some disparities, the concentration of the best performances is for $n = 2$ and $n = 3$, indicating that increasing the training set more and more does not guarantee high performances. The increase in the training set is only valid if the variability of the examples is accompanied, otherwise it can cause overfitting. In addition to the lack of performance guarantee for very large sets, derived from a few original examples, the computational cost will increase proportionally, both in image generation and during training (see more details in Section V-C).

In Table II, we have the fruits domain results (Supermarket Produce as training set and FIDS30 as test set). For this domain, the accuracy without data augmentation methods ($n = 0$) was 27.6%. Using data augmentation methods, the best performance was achieved with Correlated Noise (37.1% with $n = 2$) and the worst accuracy with Rotation (28.05%

TABLE I
ACCURACY RESULTS (%) FOR OBJECTS DOMAIN: USING AMAZON AS A SOURCE DATASET AND WEBCAM AS A TARGET DATASET. FOR THIS SETUP, THE BASELINE WAS 40.62%, I.E PERFORMANCE ACHIEVED ONLY WITH THE ORIGINAL SOURCE DATASET ON THE FINE-TUNING PROCEDURE. THE HIGHEST ACCURACY FOR EACH ROW IS HIGHLIGHTED IN BOLD.

Method	n = 1	n = 2	n = 3	n = 4	n = 5
Flip	56.85	55.72	60.00	62.64	**63.89**
Rotation	57.61	**62.76**	58.74	60.12	62.64
Shift	51.44	53.96	58.74	**66.41**	57.61
Blur	60.12	64.77	64.03	**67.92**	61.50
Correlated Noise	57.35	**68.93**	68.8	67.79	68.93
Sharpening	58.74	64.90	**66.66**	64.77	60.0
Blend	55.84	**64.52**	63.01	63.77	62.76
Threshold	61.00	66.41	**67.04**	64.40	64.15
Visual SMOTE	65.15	65.53	**66.28**	65.28	64.29

TABLE II
ACCURACY RESULTS (%) FOR FRUITS DOMAIN: USING SUPERMARKET PRODUCE AS A SOURCE DATASET AND FIDS30 AS A TARGET DATASET. FOR THIS SETUP, THE BASELINE WAS 27.60%, I.E PERFORMANCE ACHIEVED ONLY WITH THE ORIGINAL SOURCE DATASET ON THE FINE-TUNING PROCEDURE. THE HIGHEST ACCURACY FOR EACH ROW IS HIGHLIGHTED IN BOLD.

Method	n = 1	n = 2	n = 3	n = 4	n = 5
Flip	28.05	33.03	**36.65**	28.05	33.93
Rotation	31.67	31.22	**35.74**	33.93	30.76
Shift	31.22	35.29	**37.10**	28.05	27.60
Blur	30.76	**35.29**	31.22	29.41	30.76
Correlated Noise	35.74	**37.10**	32.57	34.38	33.93
Sharpening	**33.03**	32.12	28.05	32.12	29.86
Blend	28.50	**35.74**	30.31	31.22	34.84
Threshold	31.22	30.76	**33.03**	32.12	29.86
Visual SMOTE	32.12	**35.74**	31.22	29.41	28.05

with $n = 1$), representing gains from 0.45% to 9.5%. As the object domain, $n = 1$ does not provide high satisfactory results, performing better for $n = 2$ and $n = 3$. Once again, the statement that excessive generation of examples does not translate into performance gains is seen in cross-dataset scenarios. An observable behavior of the methods for this domain is the oscillation of performances when we differentiate the value of n. There is no smooth curve of growth and/or decrease, presenting successive gains and losses in this variation, except for Shift and Visual SMOTE. However, the performance difference among the distinct methods is less pronounced for this domain: Correlated Noise (37.1% for $n = 2$) and Sharpening (33.03% for $n = 1$).

To complement the view of the data augmentation performances for cross-dataset scenarios, an additional experiment was to select random examples of different techniques to fine-tune the pre-trained network. The first technique set encompasses all the methods performed here ("All"). The other techniques sets define the proposed methods, separating them through the generation of images using an example ("Single") or a pair of examples ("Combined"). The random sampling of the generated examples guarantees the same amount of images to carry out the network training. Thus, the performances can be directly compared to the results obtained for $n = 1$ and $n = 2$ from Tables I and II, as worse and better performances.

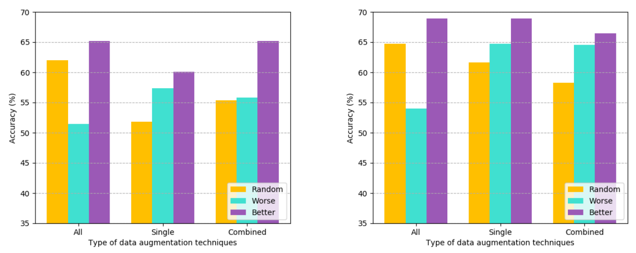

Fig. 4. Accuracies using random examples selection for objects domain: (left) Using $n = 1$; (right) Using $n = 2$.

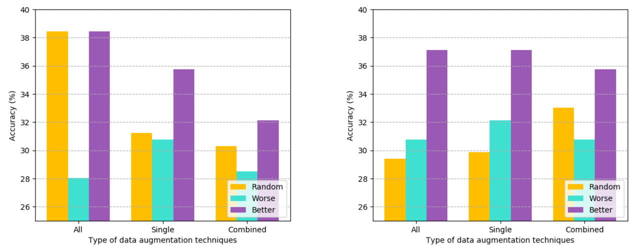

Fig. 5. Accuracies using random examples selection for fruits domain: (left) Using $n = 1$; (right) Using $n = 2$.

In Figures 4 and 5, we have the comparison for the object and fruit domains, respectively. In both results, the random selection of training examples from the data augmentation techniques provides lower accuracy than the best results obtained when directly choosing one of the methods. When compared to the worst performances, training with random examples tend to be better. However, this behavior cannot be generalized, since the oscillation occurs by varying the domain and the value of n.

B. Data augmentation for self-supervised learning

In addition to the performance analysis considering the training size set saturation, generated by the data augmentation techniques explored, we also analyzed the efficiency of self-supervised learning in cross-dataset scenarios. Thus, considering a data augmentation technique, we apply self-supervision followed by traditional fine-tuning and compared to respective fine-tuning approach only. For these experiments, we consider both domains of Office and Fruits. Additionally, we also selected a half set of 15 classes from the Office domain, called "Medium Office", chosen randomly, to check the sensitivity of self-supervision when we reduced the number of classes.

Observing the results (see Table III), we can note that the fruit domain benefits more from the self-supervision learning before conventional fine-tuning. In contrast, the office object domain does not benefit from this additional step as pre-processing. From the perspective of the number of classes, it is evident that, in classification tasks with cross-datasets, the number of classes can interfere in the sensitivity of self-supervision. Office objects have 31 classes and self-

TABLE III
ACCURACY RESULTS (%) FOR FINE-TUNING APPROACH (FT) AND SELF-SUPERVISED LEARNING FOLLOWED BY FINE-TUNING (SSFT), WHEN $n = 1$. BOLD SSFT PERFORMANCES ARE SUPERIOR TO THE RESPECTIVE PERFORMANCE WITH FT.

Method	**Office**		**Medium Office**		**Fruits**	
	FT	**SSFT**	**FT**	**SSFT**	**FT**	**SSFT**
Flip	56.85	44.77	75.00	72.55	28.05	**32.57**
Rotation	57.61	43.77	76.90	**77.17**	31.67	**33.48**
Shift	51.44	39.74	74.72	72.82	31.22	**33.48**
Blur	60.12	43.39	73.09	63.58	30.76	**33.93**
Correlated Noise	57.35	52.95	77.98	75.27	35.74	**36.19**
Sharpening	58.74	43.39	76.08	67.11	33.03	29.41
Blend	55.84	50.44	75.54	**76.9**	28.5	**35.29**
Threshold	61.00	48.80	73.36	64.13	31.22	28.05
Visual SMOTE	65.15	55.09	74.72	74.18	32.12	**36.65**
Single	51.82	52.07	77.17	61.95	31.22	**37.55**
Combined	55.34	52.45	78.26	67.11	30.31	25.79
All	62.01	49.18	82.06	66.3	38.46	28.05

supervision reduces performance for all approaches. However, this efficiency is a little more incorporated with Medium Office, containing only 15 classes, selected from the original set. The fruits domain has greatly improved with the adoption of self-supervision, being composed of only 7 classes. Consequently, there is a trend for cross-dataset classification, where fewer classes may benefit more than scenarios with a greater number of labels when self-supervised learning is applied.

Specifically, considering the data augmentation techniques for the fruit domain, we can see that Blend and Visual SMOTE, two techniques that operate image combinations, provide the greatest performance gain, 6.79% and 4.53% respectively. This performance gain is justified by the small number of classes in the dataset and the large difference among the images generated and the original ones.

C. Augmented and self-supervised computational costs

Data augmentation techniques add computational costs due to image generation and during network training. In addition to the training time, there is a cost to generate the new augmented images, which varies according to the complexity of each algorithm and the number of replications. Although we do not intend to fully investigate computational complexity, we measured the running times for the Amazon dataset with the rotation technique using the same setup of our experiments: doubling the training set size, and training it for 1000 epochs. We performed these measures using a GPU Nvidia Tesla P100 with 3584 Cuda Cores and 16GB vRAM, 2 CPUs Intel Xeon E5-2650v4 2.2 GHz with 12 kernels and 128 GB DDR3 1866MHz of RAM. To generate the augmented set (2817 images of 300×300) it was necessary a fixed time of $0.58s$, i.e. 2 milliseconds per image. Training only with the original dataset took $15.8s$/epoch on average, while for augmented training set it was necessary $32.24s$/epoch, i.e approximately 2 times more. By taking into consideration the best scenario for each dataset, improvements from 8% to 24% were obtained by doubling the training set size via augmentation. Although one has to take this into consideration, this additional cost is reasonable given the accuracy improvement. When self-supervision is included in the learning process, the processing time by epoch remains the same, since the exact setup is used in both approaches and the training data is equally the same. Therefore, when applying self-supervision followed by fine-tuning, the total time is doubled. However, as the size of the training set generated by the data augmentation techniques increases significantly, $n = 5$ for example, we realize that the trend is to reduce the classification performance. Hence, generation and training time increases significantly.

VI. GUIDELINES

When transfer learning to different datasets, alternative methods for pixel manipulation, such as Blur and Correlated Noise, tend to provide higher performances than popular approaches (such as rotation and flip) and methods that operate by mixing images. Thus, as mentioned above, these methods are highly cost-effective, since they are less computationally complex than learning or meta-learning approaches [12]. Consequently, as guidelines for applying data augmentation to cross-dataset scenarios we recommend the following observations:

a) in datasets with few classes, self-supervised learning using pseudo-label via data augmentation can be used as a warmup stage in order to improve the final classification performance;

b) data augmentation methods that perform pixel manipulation are less computationally intensive and provide good results;

c) augmenting the data has a limit in terms of being useful for training; a massive increase in training set size without a variability only causes additional learning costs and does not guarantee improved performance. Thus, if the class has few instances, performing combinations among them does not provide an increase in representation.

Our study does not cover possible combinations of techniques that may enhance the results. This restriction was imposed to ensure an individual analysis of these techniques in the context of cross-dataset, being an approach to be investigated in the future.

VII. CONCLUSIONS

In this paper, we investigated six augmentation techniques beyond the popular flip/rotation, with high potential to improve cross-dataset feature learning. We present diverse results and discussed it considering the saturation of training set size and the influence of self-supervision. We demonstrated that adding multiple data augmentation methods simultaneously or increasing the training set size is no guarantee of better performance. Instead, generating many images from the original set increases only the quantity and not the quality, especially in sets with few examples per class. Our findings shed light on transfer learning classification scenarios with small datasets, demonstrated by using difficult cross-dataset scenarios. Augmentation based on single images has great potential for this matter. Also, augmented training sets allowed faster convergence, which may be explained by the increase in the visual variety of the images. As future work, we point to the need to evaluate complementary domains other than the ones described here and the usage of unlabeled databases as data augmentation for semi-supervised learning.

VIII. Acknowledgments

We would like to thank FAPESP (grants #2019/07316-0 and #) and CNPq (National Council of Technological and Scientific Development) grant 304266/2020-5.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] S. Kornblith, J. Shlens, and Q. V. Le, "Do better imagenet models transfer better?" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2661–2671.

[7] 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.

[8] 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.

[9] 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.

[10] S. Haykin, "Neural networks: principles and practice," *Bookman*, 2001.

[11] 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*, vol. 132, pp. 131–143, 2020.

[12] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, p. 60, 2019.

[13] A. Gasparetto, D. Ressi, F. Bergamasco, M. Pistellato, L. Cosmo, M. Boschetti, E. Ursella, and A. Albarelli, "Cross-dataset data augmentation for convolutional neural networks training," in *2018 24th International Conference on Pattern Recognition (ICPR).* IEEE, 2018, pp. 910–915.

[14] L. A. Zanlorensi, E. Luz, R. Laroca, A. S. Britto, L. S. Oliveira, and D. Menotti, "The impact of preprocessing on deep representations for iris recognition on unconstrained environments," in *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI).* IEEE, 2018, pp. 289–296.

[15] B. C. Benato, A. C. Telea, and A. X. Falcão, "Semi-supervised learning with interactive label propagation guided by feature space projections," in *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI).* IEEE, 2018, pp. 392–399.

[16] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 113–123.

[17] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *arXiv preprint arXiv:1712.04621*, 2017.

[18] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," *arXiv preprint arXiv:1708.04896*, 2017.

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

[20] 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.

[21] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16.* Springer, 2020, pp. 776–794.

[22] A. Kolesnikov, X. Zhai, and L. Beyer, "Revisiting self-supervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1920–1929.

[23] L. Vogado, R. Veras, K. Aires, F. Araújo, R. Silva, M. Ponti, and J. M. R. Tavares, "Diagnosis of leukaemia in blood slides based on a fine-tuned and highly generalisable deep learning model," *Sensors*, vol. 21, no. 9, p. 2989, 2021.

[24] J. Lemley, S. Bazrafkan, and P. Corcoran, "Smart augmentation learning an optimal data augmentation strategy," *Ieee Access*, vol. 5, pp. 5858–5869, 2017.

[25] R. Takahashi, T. Matsubara, and K. Uehara, "Data augmentation using random image cropping and patching for deep cnns," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[26] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.

[27] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *arXiv preprint arXiv:1803.07728*, 2018.

[28] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with convolutional neural networks," *Advances in neural information processing systems*, vol. 27, pp. 766–774, 2014.

[29] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *European conference on computer vision.* Springer, 2016, pp. 69–84.

[30] G. S. Thumé, "Geração de imagens artificiais e quantização aplicadas a problemas de classificação," Ph.D. dissertation, Universidade de São Paulo, 2016.

[31] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271).* IEEE, 1998, pp. 839–846.

[32] D. E. Knuth, *Art of computer programming, volume 2: Seminumerical algorithms.* Addison-Wesley Professional, 2014.

[33] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[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] 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.

[36] Š. Marko, "Automatic fruit recognition using computer vision," *Mentor: Matej Kristan), Fakulteta za racunalništvo in informatiko, Univerza v Ljubljani*, 2013.

[37] W. Deng, L. Zheng, Q. Ye, G. Kang, Y. Yang, and J. Jiao, "Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 994–1003.

[38] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim, "Image to image translation for domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4500–4509.

[39] Y. Yuan, M. Chao, and Y.-C. Lo, "Automatic skin lesion segmentation using deep fully convolutional networks with jaccard distance," *IEEE Trans. Med. Imaging*, vol. 36, no. 9, pp. 1876–1886, 2017.

[40] M. Geng, Y. Wang, T. Xiang, and Y. Tian, "Deep transfer learning for person re-identification," *arXiv preprint arXiv:1611.05244*, 2016.

[41] D. Masters and C. Luschi, "Revisiting small batch training for deep neural networks," *arXiv preprint arXiv:1804.07612*, 2018.