LaSO: Label-Set Operations networks for multi-label few-shot learning

Amit Alfassy*[1,3], Leonid Karlinsky*, Amit Aides*, Joseph Shtok, Sivan Harary, Rogerio Feris
[1]IBM Research AI

Raja Giryes
[2]Tel-Aviv University

Alex M. Bronstein
[3]Technion

Abstract

Example synthesis is one of the leading methods to tackle the problem of few-shot learning, where only a small number of samples per class are available. However, current synthesis approaches only address the scenario of a single category label per image. In this work, we propose a novel technique for synthesizing samples with multiple labels for the (yet unhandled) multi-label few-shot classification scenario. We propose to combine pairs of given examples in feature space, so that the resulting synthesized feature vectors will correspond to examples whose label sets are obtained through certain set operations on the label sets of the corresponding input pairs. Thus, our method is capable of producing a sample containing the intersection, union or set-difference of labels present in two input samples. As we show, these set operations generalize to labels unseen during training. This enables performing augmentation on examples of novel categories, thus, facilitating multi-label few-shot classifier learning. We conduct numerous experiments showing promising results for the label-set manipulation capabilities of the proposed approach, both directly (using the classification and retrieval metrics), and in the context of performing data augmentation for multi-label few-shot learning. We propose a benchmark for this new and challenging task and show that our method compares favorably to all the common baselines.

1. Introduction

Deep learning excels in creating informative and discriminative feature spaces for many types of data, e.g. natural images [13, 14, 17]. In modern computer vision, image representation in a deep feature space is expected to encode all of the semantic content of interest, whether it is the object categories present in the image [14], their visual attributes [9], or their locations [13]. Usually, these feature spaces are trained using large quantities of labeled data tailored to the task [20, 31]. However, in many practical applications, only a handful of examples are available for the target task; this scenario is known as few-shot learning [37].

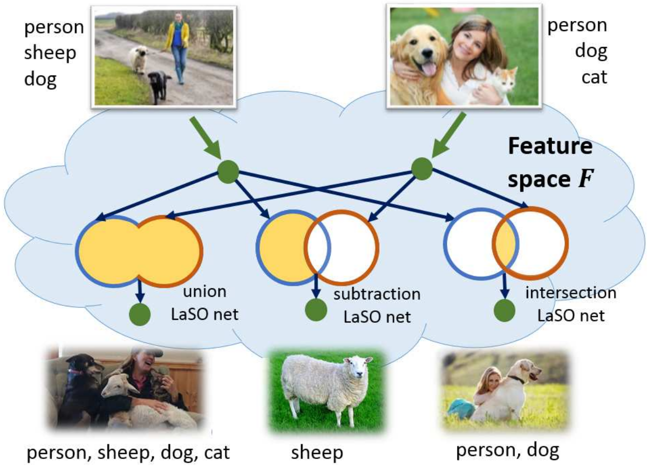


Figure 1. LaSO networks operating in a feature space. The goal of these networks is to synthesize new feature vectors from pairs of input vectors so that the semantic content of the synthesized vector matches the prescribed operation on the source vector's label sets.

In few-shot learning, the feature spaces are usually transferred from other tasks, either directly or through meta-learning that allows generating these spaces on the fly (see survey of such techniques in Section 2). One popular approach for few-shot learning is the generative one [12, 25, 32, 44]: Many new examples in the chosen feature space are generated from the few given training examples; these synthesized samples are in turn used to improve the generalization of the few-shot task. Despite the increasing popularity of few-shot learning, all the current works on few-shot classification deal with a single (class) label per data point (e.g. $C(Img) = dog$), and not with the multi-label case (e.g. $C(Img) = \{dog, leash, person, forest\}$).

In this paper, we propose a new kind of a generative approach to few-shot learning. It explicitly targets multi-label samples; even more so, through its task definition, it targets cases where the labels are not necessarily explicitly defined a-priori. As an illustrative example, please consider the situation depicted in Figure 2. Suppose you wish to build a

*The authors have contributed equally to this work

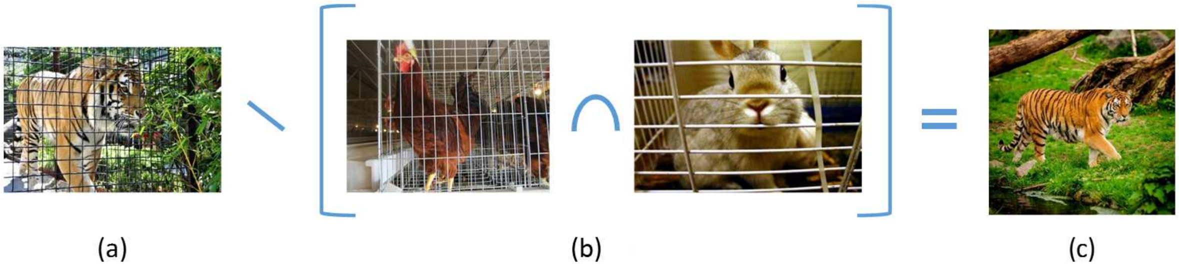


Figure 2. **LaSO concept:** manipulating the semantic content of the (small) data for better generalization to situations beyond what was originally observed. The manipulation is based on the data itself and is performed in *feature space*. For real examples of our approach performing the $A \setminus (B \cap C)$ operation on real images, please see figure 4d.

(multi-label) classifier for wild animals. You go to a zoo and take a few photos of each animal (so the learning task is a *few-shot*). But alas, all of the animals are caged (Figure 2(a)) and this few-shot trained classifier is likely to have some difficulty with the generalization to animals in the wild (Figure 2(c)). Note that in this case, the label 'caged' is not even part of the label vocabulary used for the manual annotation (here the vocabulary only contains animals).

To address this issue, we propose having neural networks that can manipulate the 'semantic content' of the samples in feature space 'by example' (e.g. suppress in a feature vector elements corresponding to labels that correspond to another feature vector). For instance, consider having a model, M_{int}, that can accept two images with caged animals in some feature space (Figure 2(b)), and produce a feature vector representing their common semantic content. Since the shared (implicit) concept here is the 'cage', it should end up with a feature vector representing 'caged' (that is if we had a classifier for 'caged' it would fire on this vector), but no longer representing either of the caged animals appearing in the original intersected images (rooster and a rabbit in this case). Then consider having another model, M_{sub}, that can implicitly remove concepts present in one sample from another sample (again in feature space). We can then apply M_{sub} on the caged tiger and the feature vector representing 'caged' that we obtained using M_{int}, thus effectively getting a feature vector for a 'tiger in the wild'. Figure 4(d) shows examples of the $A \setminus (B \cap C)$ operation performed on real images using our approach.

Equipped with this concept, we propose to build and train a complete set of sample-based content manipulation models in feature space, namely M_{int} for the label set *intersection* operation, M_{uni} for the label set *union*, and M_{sub} for the label set *subtraction*. We call these models Label Set Operations networks (or LaSO nets for short). A schematic illustration is given in Figure 1. The pair of images, entering the system, are converted to feature vectors using some backbone network and then processed by any of the aforementioned manipulation networks to produce feature vectors with corresponding label sets.

In Section 4 (results), we show that our approach exhibits an ability to generalize to unseen (unlabeled) concepts allowing us to apply the LaSO nets to semantic concepts not present in the set of previously observed labels (like the label 'caged' in the previous example). In our experiments, LaSO nets demonstrate a far from chance level of success in manipulating labels unseen during training. This in turn allows our approach to be applied in the multi-label few-shot scenario, generating synthetic examples by manipulating novel classes unseen during training.

To summarize, our main contributions are threefold. **First**, we propose a method for the *few-shot multi-label* learning task, a novel direction in few-shot learning research, not addressed so far in the literature. **Second**, we propose a novel concept of *by-example label-set manipulation* in feature space, allowing the generation of new multi-label samples by combining other samples. In our approach the manipulation on the labels of the combined samples is defined by the semantic content of the samples themselves and hence does not necessarily require an explicit supervised pre-training of all possible desired manipulations. **Third**, we offer the community the first benchmark for the few-shot multi-label learning task, accompanied with a set of performance evaluations and baseline comparisons.

This paper is organized as follows. Section 2 reviews related work in the fields of few-shot learning and training samples augmentation. Section 3 explains the technical details of our proposed approach. Section 4 reviews the various experiments and results. Finally, Section 5 presents our conclusions and suggestions for future work.

2. Related Work

Recently, the problem of few-shot learning has received much attention in the computer vision community. In the Meta-Learning (or learning-to-learn) approach [10, 19, 23, 29, 33, 37, 43], classification models are trained not on individual annotated samples, but rather on instances of the few-shot learning tasks, comprised of a small training set and a number of query samples. The goal of meta-learning is to learn a model that produces models for any such few-shot task, usually without (or with only a short) fine-tuning.

Another line of works in few-shot learning is characterized by enriching the small initial training dataset using data augmentation and data synthesis techniques. Simple image transformations (horizontal flips, scaling, shifts), have been exploited in the machine learning community from the beginning. The work in [28] takes this type of augmentation to the next level by learning a sequences of user-defined (black-box) transformations, along with their parameters, that keep the objects recognizable.

In the synthesis approaches, new examples are generated based on the few provided labeled ones (in out-of-sample manner). Some works render synthetic examples using geometric deformations [25] or CNNs [7, 34]; specifically, a strong recent trend is to generate examples using Generative Adversarial Networks (GANs) [8, 11, 15, 16, 22, 27, 30, 44]. In other works, the example synthesis is done using additional semantic information [4, 41], relative linear offsets between elements of the same category in feature space [12], learning to extract and apply a non-linear transformation between pairs of examples of the same category [32], or training augmentation and classification modules end-to-end in a closed loop [39].

The approach for sample synthesis taken in this work relies on generating new samples corresponding, on the level of semantic labels, to intersection, union or subtraction of the labels present in two input samples. These labels may be objects or attributes that are present in the input samples. The set operations are non-degenerate only in the *multi-label* scenario, either when each image contains multiple objects (e.g. MS-COCO dataset) or a single objects with multiple attributes (e.g., CelebA dataset).

Some prior works on multi-label classification improve upon the straightforward approach of having an independent classifier per label by learning label correlations within images (see [38] for an extensive review). Yet, in the few-shot domain, this information cannot be exploited for a new task, which contains unseen categories. In [2], the task of few-shot multi-label text classification is addressed, relying on the structure of the label space specific to text. In [18] multi-label zero-shot classification is explored and in [40] rare structured data combinations are being augmented by external data crawling. To the best of our knowledge, there is no prior work of multi-label few-shot visual categories classification without using external data.

In the domain of object composition, [24] models attributes as operators, learning a semantic embedding that explicitly factors out attributes from their accompanying objects, in order to recognize unseen attribute-object compositions. In [3], a pipeline for integrating two visual objects is proposed, for the purpose of generating images composed of the two objects, spatially combined (tested on synthetic data). This task is very different than the one we would like to address, as: (1) a spatial combination of objects requires to learn occlusions; and (2) the composition takes place in the image space, rather than on the feature level, as in our approach. The latter provides the ability to use existing feature extractors (such as Inception [35] or ResNet [14]) more easily, which is more applicable for few-shot classification.

3. Method

Our approach is schematically illustrated in Figure 3. Input images X and Y, each with a corresponding set of multiple labels, $L(X), L(Y) \subseteq \mathcal{L}$ respectively, are represented in the joint feature space $\mathcal{F}$ as F_X and F_Y. This space $\mathcal{F}$ is realized using a backbone feature extractor network $\mathcal{B}$; we have used InceptionV3 [35] and ResNet-34 [14] backbones in our experiments. Three LaSO networks M_{int}, M_{uni}, and M_{sub} receive the concatenated F_X and F_Y and are trained to synthesize feature vectors in the same space $\mathcal{F}$. As the name (*int=intersection*) suggests, M_{int}'s goal is to synthesize a feature vector

$$M_{int}\left(F_X, F_Y\right) = Z_{int} \in \mathcal{F} \tag{1}$$

which corresponds to a hypothetical image I, such that $\mathcal{B}(I) = Z_{int}$ and $L(I) = L(X) \cap L(Y)$. In other words, this means that if a human would observe and label I, it would receive $L(X) \cap L(Y)$ as its label set. Similarly, M_{uni} and M_{sub} output $Z_{uni}, Z_{sub} \in \mathcal{F}$ that are expected to correspond to the union of the label sets $L(X) \cup L(Y)$, and the subtraction of the label sets $L(X) \setminus L(Y)$ respectively.

Note that although we use a pre-defined set of labels $\mathcal{L}$ for training our models, we can expect that during training, the networks will also generalize to labels which are not part of $\mathcal{L}$. This is possible because LaSO nets receive no explicit label information as input (neither during training, nor during use). They are forced to learn to synthesize vectors corresponding to the desired label sets implicitly, only by observing F_X and F_Y as their inputs, without being explicitly given their labels. In Section 4 (Results) we test this ability of our networks to generalize to novel categories.

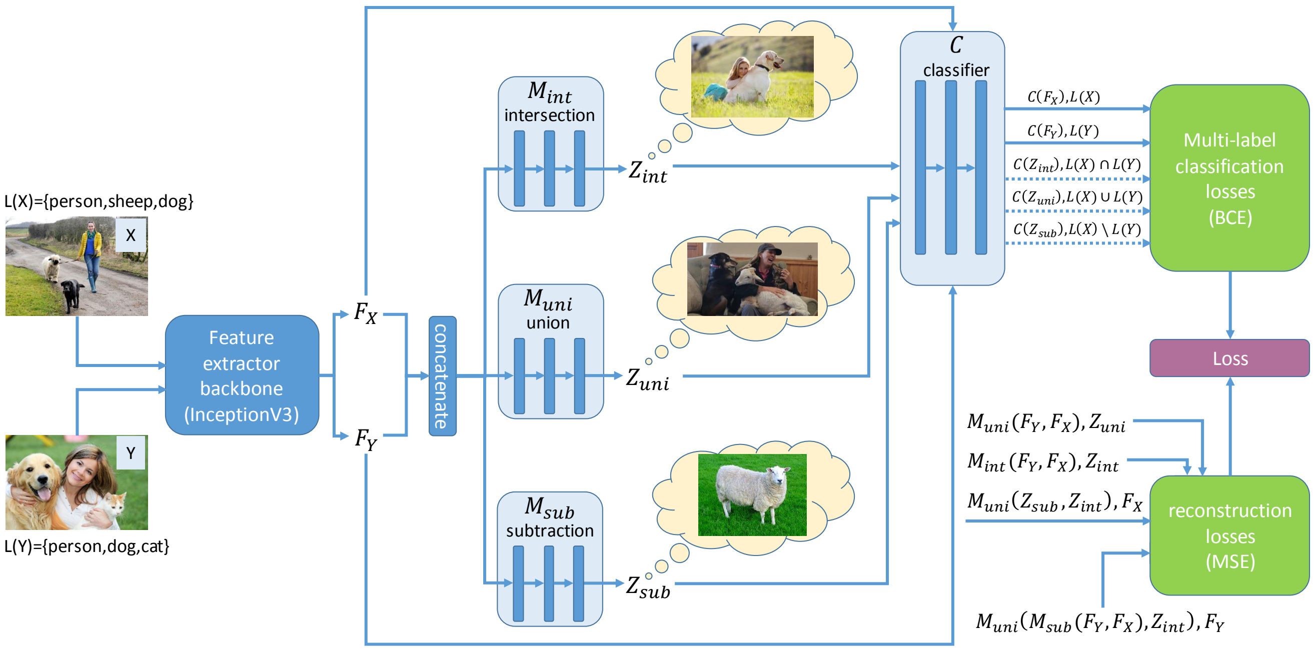
The source feature vectors, F_X and F_Y, and the outputs of the LaSO networks, namely Z_{int}, Z_{uni}, and Z_{sub}, are fed into a classifier C. We use the Binary Cross-Entropy (BCE, aka Sigmoid-Cross-Entropy) multi-label classification loss in order to train C and the LaSO networks:

$$BCE(s, l) = -\sum_i l_i \log \sigma(s_i) + (1 - l_i) \log(1 - \sigma(s_i)) \tag{2}$$

with the sigmoid $\sigma(x) = (1+\exp(x))^{-1}$, the vector s being the classifier scores, l being the desired (binary) labels vector, and i the class indices. To train the classifier C we use only the combination of the losses from the source feature vectors:

$$C_{loss} = BCE(C(F_X), L(X)) + BCE(C(F_Y), L(Y)) \tag{3}$$

where $C(\cdot)$ stands for the classifier C output score vector. The LaSO networks are trained using:



L(X)={person,sheep,dog}
X
L(Y)={person,dog,cat}
Y
Feature extractor backbone (InceptionV3)
F_X
F_Y
concatenate
M_{int} intersection
M_{uni} union
M_{sub} subtraction
Z_{int}
Z_{uni}
Z_{sub}
C classifier
$C(F_X), L(X)$
$C(F_Y), L(Y)$
$C(Z_{int}), L(X) \cap L(Y)$
$C(Z_{uni}), L(X) \cup L(Y)$
$C(Z_{sub}), L(X) \setminus L(Y)$
Multi-label classification losses (BCE)
Loss
$M_{uni}(F_Y, F_X), Z_{uni}$
$M_{int}(F_Y, F_X), Z_{int}$
$M_{uni}(Z_{sub}, Z_{int}), F_X$
reconstruction losses (MSE)
$M_{uni}(M_{sub}(F_Y, F_X), Z_{int}), F_Y$

Figure 3. **LaSO model:** schematic illustration of all the components of the proposed approach (including training losses).

$$\begin{aligned} LaSO_{loss} = & BCE(C(Z_{int}), L(X) \cap L(Y)) + \\ & BCE(C(Z_{uni}), L(X) \cup L(Y)) + \\ & BCE(C(Z_{sub}), L(X) \setminus L(Y)) \end{aligned} \tag{4}$$

For the LaSO updates the classifier C is kept fixed and only used for passing gradients backwards. Note that the used losses decouple the training of C and the LaSO networks.

In addition, our model includes a set of Mean Square Error (MSE) based reconstruction losses. The first loss is used to enforce symmetry for the symmetric *intersection* and *union* operations. This loss R^{sym}_{loss}, is realized as the MSE between $Z_{int} = M_{int}(F_X, F_Y)$, $Z_{uni} = M_{uni}(F_X, F_Y)$ and the vectors obtained from the corresponding networks with the reversed order of the inputs:

$$\begin{aligned} R^{sym}_{loss} = & \frac{1}{n} \| Z_{int} - M_{int}(F_Y, F_X) \|_2 + \\ & \frac{1}{n} \| Z_{uni} - M_{uni}(F_Y, F_X) \|_2 \end{aligned} \tag{5}$$

Please note that $M_{int}(F_X, F_Y)$ and $M_{int}(F_Y, F_X)$ invoke the same instance of M_{int}. Same holds for any LaSO network that is invoked multiple times in our construction.

The second loss is used in order to reduce the chance of mode collapse that could cause a semi-fixed output for each possible label set combination. For example, in case of a mode collapse, we could observe very similar outputs of the network M_{int} for many different pairs of images with the same set of shared labels. The mode collapse related reconstruction loss, R^{mc}_{loss}, is realized as the MSE loss between F_X, F_Y and the outputs of simple expressions (generated by some combinations of the LaSO networks) that produce feature vectors that should correspond to the original label sets $L(X)$ and $L(Y)$ by set-theoretic considerations:

$$\begin{aligned} R^{mc}_{loss} = & \frac{1}{n} \| F_X - M_{uni}(Z_{sub}, Z_{int}) \|_2^2 + \\ & \frac{1}{n} \| F_Y - M_{uni}(M_{sub}(F_Y, F_X), Z_{int}) \|_2^2, \end{aligned} \tag{6}$$

where n is the length of F_X.

3.1. Implementation details

We have implemented our approach using PyTorch 1.0 [26]. The InceptionV3 and the ResNet-34 feature extractor backbones are pre-trained from scratch using the corresponding training sets as described in Section 4 (Results). The LaSO networks are implemented as Multi-Layer Perceptrons (MLPs) consisting of 3 or 4 blocks. Each block contains a fully-connected layer followed by batch-normalization, leaky-ReLU, and dropout. A future work may explore additional architectures for the LaSO nets, e.g. encoder-decoder and residual based architectures. During training, we used batch size of 16, initial learning rate of 0.001, learning rate reduced on loss plateau with factor 0.3. The optimization is performed with the Adam optimizer [6] with parameters $(0.9, 0.999)$. The code is available here.

4. Results

An image usually contains multiple object instances that can be translated to a set of unique category labels. Object detection and segmentation datasets are a great source of multi-object labels. Indeed, by throwing away the bounding boxes and segmentation masks, and keeping only the unique category labels set we can transform any such dataset into a

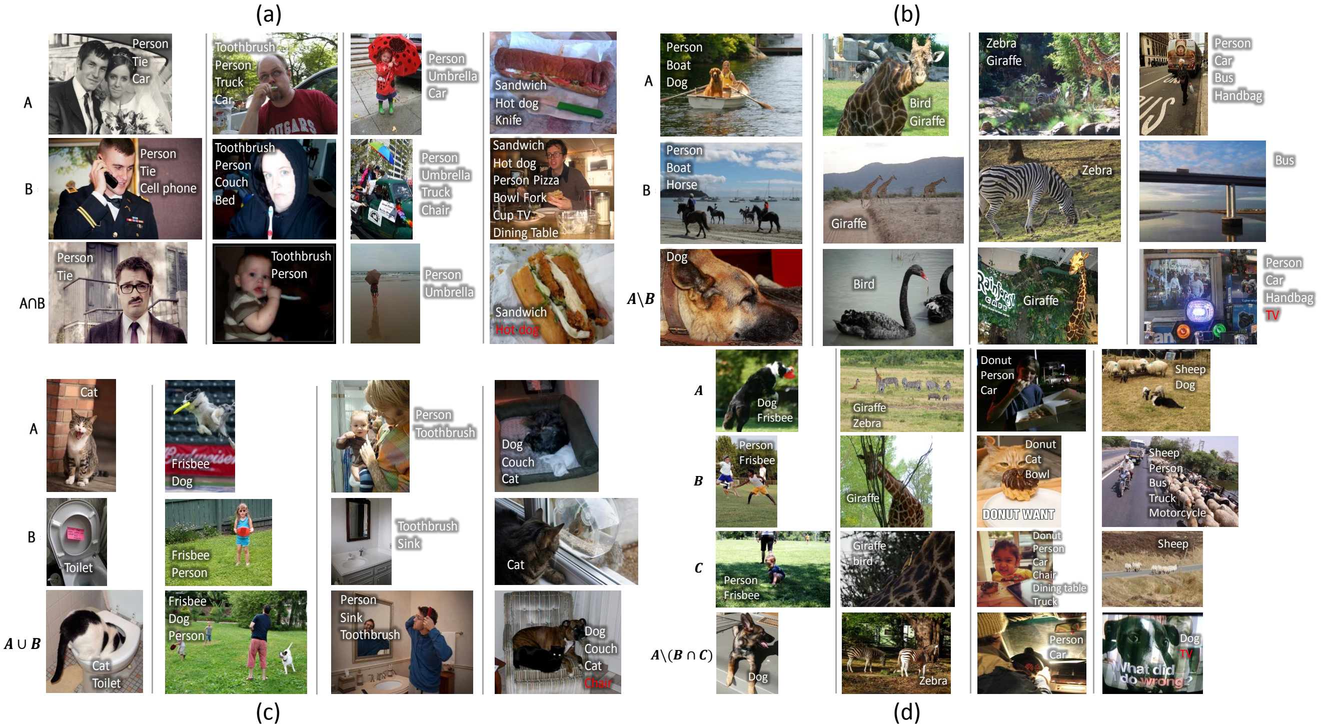


Figure 4. **Testing LaSO networks using retrieval:** A and B feature vectors are inputs to LaSO nets and the nearest neighbor image in feature space to the output feature vector is shown below each pair. For each operation we show three successful examples and one failure case highlighting the errornous label in red. Best viewed in color. (a) intersection retrieval examples; (b) subtraction retrieval examples; (c) union retrieval examples; (d) $A \setminus B \cap C$ retrieval examples.

multi-label classification one. In our experiments we used the popular (and challenging) MS-COCO [20] dataset as the source of multi-object labels.

An object, e.g. a face, can be described in terms of its various attribute labels. To test our approach on the task of manipulating the attribute-based multi-label data, we have used the CelebA [21] dataset. In CelebA experiments we have used its 40 facial attribute annotations as labels.

4.1. MS-COCO experiments

For MS-COCO experiments we have used the COCO 2014 train and validation sets. The 80 COCO categories were randomly split into 64 'seen' and 16 'unseen' categories. The unseen categories were: *bicycle, boat, stop sign, bird, backpack, frisbee, snowboard, surfboard, cup, fork, spoon, broccoli, chair, keyboard, microwave*, and *vase*. We filtered the COCO train set leaving only images that did not contain any of the 16 unseen category labels and used this filtered set to train our feature extractor backbone (InceptionV3) and the LaSO models (as described in section 3). Before training jointly with the LaSO models, the feature extractor backbone was first pre-trained separately as a multi-label classifier for the 64 seen categories on the filtered training set using the standard BCE classification loss.

	64 seen classes	16 unseen classes
intersection	77	48
union	81	61
subtraction	43	14
non-manipulated feature vectors	75	79

Table 1. Evaluating feature vectors synthesized by the LaSO networks using the *classification* performance (in mAP %) on the 64 *seen* and on the 16 *unseen* COCO categories on COCO validation set. Classification is performed w.r.t. the expected label set after each type of operation. Performance on the original feature vectors is given as measure of feature space capacity for classification.

4.1.1 Evaluating the label set manipulation capability of the LaSO networks

We used the COCO validation set to test the performance of the LaSO models for the label set intersection, union, and subtraction operations. We used two methods for this evaluation, one using classification and the other using retrieval.

In the classification tests we have used a classifier pretrained on the feature space $\mathcal{F}$ (generated by the backbone feature extractor model) to test the LaSO networks. To this end, we have randomly paired all of the validation set images and tested each LaSO operation network on each pair. For any pair of images X and Y, and their corresponding feature vectors F_X and F_Y, the outcome of $M_o(F_X, F_Y)$, where $o \in \{uni, int, sub\}$, was fed to the classifier and its

	64 seen classes			16 unseen classes		
	top-1	top-3	top-5	top-1	top-3	top-5
intersection	0.7	0.79	0.82	0.47	0.71	0.78
union	0.61	0.71	0.74	0.44	0.64	0.71
subtraction	0.19	0.32	0.4	0.21	0.4	0.51
non-manipulated feature vectors (upper bound)	0.56	0.72	0.76	0.56	0.75	0.81

Table 2. Evaluating feature vectors synthesized by the LaSO networks using the *retrieval* performance on the 64 *seen* and on the 16 *unseen* MS-COCO categories (Sec. 4.1.1). Retrieval quality is measured w.r.t. the expected label set after each type of operation. All tests are performed on the MS-COCO validation set, not used for training. Numbers are *mean Intersection over Union* (mIoU) between the label sets of the retrieved samples and the expected label set, the mean is taken over the different queries. The top-k averages the maximum IoU obtained among closest k retrieved samples. The retrieval performance of the non-manipulated feature vectors measures the capacity of the feature space $\mathcal{F}$ for performing retrieval, and hence provide an upper bound on the LaSO retrieval performance.

resulting class scores were evaluated vs the expected label-set resulting from applying the set operation o on $L(X)$ and $L(Y)$. We performed two separate evaluations, one for the seen and the other for the unseen categories. In each of the tests we compute the Average Precision (AP) for each category and report the mean AP (mAP) computed over the categories in each (seen / unseen) set.

The 64-way classifier that was used for LaSO evaluation on the 64 seen categories was the one that was trained together with the backbone model $\mathcal{B}$ (that generate the feature space $\mathcal{F}$). The 16-way classifier that was used for the 16 unseen categories evaluation was trained on those images from the COCO training set that contained instances of these 16 categories. For its training, we used the same feature space $\mathcal{F}$ generated by our (frozen) backbone $\mathcal{B}$ (trained in the course of LaSO training). The reason is that the trained LaSO networks can only operate in this space. The results of the classification based evaluation experiments are summarized in Table 1. On the set of seen categories, for the union and intersection operations, the LaSO networks managed to learn to synthesize feature vectors which through the eyes of the classifier are seen as comparable (even slightly better) to the original non-manipulated feature vectors. On the unseen categories there is still room for improvement. Yet even there the results are well above chance, indicating that despite not observing any of the unseen categories during training, the LaSO label set manipulation operations managed to generalize beyond the original training labels. This opens the door for the multi-label few-shot experiments on the set of the unseen categories presented in section 4.1.4 below.

In the retrieval tests we have evaluated the synthesized feature vectors directly without using any classifier. We used nearest neighbor search in a large pool of feature vectors of real images with ground truth labels. To this end, as in the classification tests, validation images were randomly paired and passed through the LaSO networks resulting in synthesized feature vectors with an expected set of labels (according to the operation). The synthesized feature vectors were then used to retrieve the first k nearest neighbors (NNs) in the validation set. Please see Fig. 4 for some examples of inputs to different LaSO nets, and the corresponding retrieved NNs. For each of the resulting NNs, Intersection over Union (IoU) was computed between the ground truth label-set of the NN and the expected label-set of the synthesized vector. Then maximum IoU was computed on the top-k NNs. In Table 2 we report average IoU computed over the entire set of the synthesized vectors, for different $k \in \{1, 3, 5\}$ and for the seen and unseen sets of categories separately. As an upper bound set by feature space $\mathcal{F}$ capacity for retrieval, we also repeated the retrieval evaluation as above for the original non-manipulated feature vectors. Again, as can be seen from the results, in terms of retrieval, the feature vectors synthesized by the LaSO networks for the intersection and the union operations are performing on par with the original non-manipulated ones. The performance is slightly better for some of the k on the set of seen categories, and quite close on the unseen ones. This again provides evidence for the ability of the LaSO networks to generalize to unseen categories and supports their use for performing augmentation synthesis for few-shot multi-label training (Sec. 4.1.4).

4.1.2 Analytic approximations to set operations

Using the (naive) interpretation of the feature vectors in the space $\mathcal{F}$ as collections of individual features correlated with the appearance of specific visual labels, we can consider analytic operations on pairs of feature vectors which mimic the effects of the set operations in the label space. This enables a simpler version of our method, which does not involve learned LaSO networks, but still generates synthetic features that can contribute to multi-label few-shot classifier training as will be demonstrated in section 4.1.4.

Denoting the input to LaSO networks by $F_X, F_Y \in \mathcal{F}$, as in the Fig. 3, we have defined and evaluated the following set of analytic LaSO alternatives:

Operator	Expression 1	Expression 2
Union	$F_X + F_Y$	$\max(F_X, F_Y)$
Intersection	$F_X \cdot F_Y$	$\min(F_X, F_Y)$
Subtraction	$F_X - F_Y$	$\mathrm{ReLU}(F_X - F_Y)$

We defined this set of alternatives drawing intuition from the DCGAN paper [27], that has proposed GAN arithmetics as an interesting possibility of manipulating images in the space of GAN random seeds. In our case, we are not assuming a (well) trained GAN for our multi-label data, and explore a simpler variant, directly manipulating feature vectors in $\mathcal{F}$. Table 4 summarizes the comparison between the top performing analytic and the learned LaSO variants on both the COCO and the CelebA datasets. In both experiments, the top performing analytic expressions were $\max(F_X, F_Y)$ for the union, $\min(F_X, F_Y)$ for the intersection, and $ReLU(F_X - F_Y)$ for the subtraction. As can be seen, the learned LaSO networks outperform the simpler analytic alternatives in almost all cases, yet in some cases the analytic versions are not far behind, indicating them as additional good candidates for being used for augmentation synthesis in few-shot multi-label experiments in section 4.1.4.

dataset	method	subtraction	intersection	union
MS-COCO	analytic	29.0	74.7	76.5
	learned	**43.0**	**77.0**	**81.0**
CelebA	analytic	37.0	**52.0**	47.0
	learned	**69.0**	48.0	**75**

Table 3. Comparing the learned operators with analytic alternatives. All numbers are in mAP %.

4.1.3 Ablation study of the reconstruction losses

To test the effect of using the different reconstruction losses used in the LaSO training (R^*_{loss}) we have run ablation experiments on the 'seen' classification mAP metric (table 4). We found that adding both reconstruction losses gives $3\% - 9\%$ gains to all operations, improving by 6% on average. In addition, to verify the utility of R^{mc}_{loss} in improving the diversity of the LaSO operations output, we have examined the diversity of samples retrieved with LaSO ops trained with and without R^{mc}_{loss}. It turned out that training with R^{mc}_{loss} improves the retrieved samples diversity by 5% on seen categories and 9% on unseen categories (on average over the 3 LaSO operations). These gains are consistent in two diversity measures computed on samples retrieved by a set of 30,000 queries: (i) total amount of retrieved ***unique*** samples; (ii) average number of repetitions per sample.

	inter.	union	subtr.	average
without R^{sym}_{loss} and R^{mc}_{loss}	0.72	0.78	0.34	0.61
+ R^{sym}_{loss}	0.76	0.81	0.39	0.65
+ R^{mc}_{loss} (final model)	0.77	0.81	0.43	0.67

Table 4. Ablation experiments for the reconstruction losses using classification mAP, evaluated on 64 'seen' COCO categories.

4.1.4 Multi-label few-shot classification experiments

In this section we explore an interesting application of the label-set manipulation concept - serving as a (learned) augmentation synthesis method for training a multi-label few-shot classifier. As opposed to the well-studied single-label few-shot classification, in the multi-label few-shot scenario the examples of different categories are only provided in groups. This renders the existing techniques for few-shot classification inapplicable, and to the best of our knowledge, this problem was not addressed before.

Therefore, we propose our own benchmark and a first set of results for this problem, comparing our approach to multiple natural baselines. The baselines are: (A) training directly on the small labeled set, (B) using basic image augmentation while training on the small labeled set, and (C) using the mixUp [42] augmentation technique. We compared these baselines to both the learned LaSO networks and the analytical alternatives discussed in Section 4.1.2.

	1-shot	5-shot
B1: no augmentation	39.2	49.4
B2: basic aug.	39.2	52.7
B3: mixUP aug.	40.2	54.0
analytic intersection aug.	40.7	55.4
analytic union aug.	44.5	55.6
learned subtraction aug.	40.0	54.1
learned intersection aug.	40.5	57.2
learned union aug.	**45.3**	**58.1**

Table 5. Multi-label few-shot mAP (in %) on 16 unseen categories from MS-COCO. The feature extractor and the LaSO networks are trained on the remaining 64 MS-COCO categories. Average of 10 runs are reported, tested on the entire MS-COCO test set. MixUP baseline uses the original code of [42].

As our benchmark, we propose the set of the 16 COCO categories unseen during training. We generate 10 random episodes (few-shot train set selection) for each of the 1-shot (1 example per category) and 5-shot (5 examples per category) scenarios. The same episodes are used for all the methods: the LaSO variants and all the baselines. During episode construction we maintained a histogram of the label counts ensuring that a total of 1 example per category appears in the episode for 1-shot scenario and 5 examples in 5-shot scenario respectively. Of course due to the random nature of the episodes, this balancing is not always possible, and hence in some episodes the amount of labels per category could exceed 1 or 5 (just by 1 in the majority of the cases). But since same exact episodes are used for all the compared approaches the comparison are fair. The entire COCO validation set (considering only the 16 unseen categories annotations) is used for testing the classifiers trained on each of the episodes.

All the training and the validation images were converted to the same feature space $\mathcal{F}$ created by our feature extraction backbone, the training and the augmentation were performed on top of $\mathcal{F}$ (except for the standard augmentation that was applied to the images and then converted to $\mathcal{F}$ by the backbone). Random pairs of examples from the small (1 or 5-shot × 16 categories) training set were used for label-set manipulations. For all the augmentation baselines and all variants of our method, same number of samples were synthesized per training epoch. On all compared approaches the classifiers trained on each of the episodes were trained using 40 SGD epochs (as we experimentally verified, all of them converged before 40 epochs).

The results of this experiment are reported in Table 5. All results are reported in mAP $\%$ computed over the 16 unseen categories in the entire COCO validation set. As can be seen from the results, for both 1 and 5 shot scenarios label set manipulation obtains stable gains of 5.1 and 4.1 mAP points respectively. This points towards the ability of the LaSO networks to generalize to unseen labels, also showing the general utility of our label-set manipulation approach in learning to augment data for training multi-label few-shot classifiers in a challenging realistic scenario (COCO).

4.2. CelebA experiments

We used the CelebA dataset [21] in order to test our approach on a different kind of multi-label data, namely object attributes. The CelebA dataset contains $\sim$ 200K images labeled according to 40 facial attributes. We pre-trained the feature extractor backbone (based on the ResNet-34) as a multi-label classifier on the training samples of the CelebA dataset. Then we trained M_{uni}, M_{int} and M_{sub} to perform the corresponding set-operations on the attribute-based multi-labels on the same training data. We repeated the classification based evaluation experiments and ablation studies as described for COCO in section 4.1. The test samples of the CelebA dataset were used to evaluate the performance. The results of the classification based evaluation are summarized in Table 6 in mAP $\%$ computed over the 40 attributes of CelebA. The union and subtraction LaSO networks achieve relatively high mAP while the intersection network scores lower. This can be attributed to the fact that the intersection network training is unbalanced and biased toward negative attributes (it leaves most attributes turned off), while the precision computation is more affected by the ability to accurately predict the positive labels. Results of the ablation studies are given in Table 4.

5. Summary & Conclusions

In this paper we have presented the label set manipulation concept and have demonstrated its utility for a new and challenging task of the multi-label few-shot classification. Our results show that label set manipulation holds a good potential for this and potentially other interesting applications, and we hope that this paper will convince more researchers to look into this interesting problem.

	40 facial attributes
intersection	48
union	75
subtraction	69
original (non-manipulated) feature vectors	79

Table 6. Evaluating feature vectors synthesized by the LaSO networks using the *classification* performance on the 40 facial attributes in CelebA. Classification is performed w.r.t. the expected label set after each type of operation, and on the original feature vectors for reference. All tests are performed on the CelebA test set, not used for training. Numbers are in mAP $\%$.

Natural images are inherently multi-label. We have focused on two major sources of labels: objects and attributes. Yet, other possible sources of image labels, such as the background context, object actions, interactions and relations, etc., may be further explored in a future work.

Another interesting future direction is exploring additional architectures for the proposed LaSO networks. For example an encoder-decoder architecture, where the encoder and the decoder subnets are shared between the LaSO networks, and the label-set operations themselves are implemented between the encoder and the decoder via the analytic expressions proposed in section 4.1.2. This alternative architecture has the potential to disentangle the feature space into a basis of independent constituents related to independent labels facilitating the easier use of analytic variants in such a disentangled space. Another interesting future direction is to use the proposed techniques in the context of few-shot multi-label semi-supervised learning, where a large scale unlabeled data is available, and the proposed approach could be used for automatic retrieval of more auto-labeled examples with arbitrarily mixed label sets (obtained by mixing the few provided examples). In addition, the proposed approach might also prove useful for the interesting visual dialog use case, where the user can manipulate the returned query results by pointing out or showing visual examples of what she/he likes or doesn't like.

Finally, the approach proposed in this work is related to an important issue in Machine Learning, known as *dataset bias* [36] or *out-of-context* recognition [1, 5]. An interesting future direction for our approach is to help reducing the bias dictated by the specific provided set of images by enabling a better control over the content of the samples.

Acknowledgments: This research was partially supported by ERC-StG SPADE PI Giryes, ERC-StG RAPID PI Bronstein, and the European Unions Horizon 2020 research and innovation programme under grant agreement 688930. Rogerio Feris is partly supported by IARPA via DOI/IBC contract number D17PC00341. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government).

References

[1] John K. Tsotsos Amir Rosenfeld, Richard Zemel. Elephant in the room. arXiv:1808.03305, 2018. 8

[2] Ramakanth Kavuluru Anthony Rios. Few-Shot and Zero-Shot Multi-Label Learning for Structured Label Spaces. EMNLP, pages 1–10, 2018. 3

[3] Samaneh Azadi, Deepak Pathak, Sayna Ebrahimi, and Trevor Darrell. Compositional GAN: Learning Conditional Image Composition. arXiv:1807.07560, 2018. 3

[4] Zitian Chen, Yanwei Fu, Yinda Zhang, Yu-Gang Jiang, Xiangyang Xue, and Leonid Sigal. Semantic Feature Augmentation in Few-shot Learning. arXiv:1804.05298v2, 2018. 3

[5] Myung Jin Choi, Antonio Torralba, and Alan S. Willsky. Context models and out-of-context objects. Pattern Recognition Letters, 33(7):853–862, 2012. 8

[6] Jimmy Ba Diederik P. Kingma. Adam: a method for stochastic optimization. 3rd International Conference for Learning Representations, 2015. 4

[7] Alexey Dosovitskiy, Jost Tobias Springenberg, Maxim Tatarchenko, and Thomas Brox. Learning to Generate Chairs, Tables and Cars with Convolutional Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4):692–705, 2017. 3

[8] Ishan Durugkar, Ian Gemp, and Sridhar Mahadevan. Generative Multi-Adversarial Networks. International Conference on Learning Representations (ICLR), pages 1–14, 2017. 3

[9] V Ferrari and A Zisserman. Learning Visual Attributes. Nips, 2007. 1

[10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. arXiv:1703.03400, 2017. 2

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. Advances in Neural Information Processing Systems 27, pages 2672–2680, 2014. 3

[12] Bharath Hariharan and Ross Girshick. Low-shot Visual Recognition by Shrinking and Hallucinating Features. IEEE International Conference on Computer Vision (ICCV), 2017. 1, 3

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. arXiv:1703.06870, 2017. 1

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. arXiv:1512.03385, 2015. 1, 3

[15] Gao Huang, Zhuang Liu, L v. d. Maaten, and Kilian Q Weinberger. Densely Connected Convolutional Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2261–2269, 2017. 3

[16] Eli Shechtman Jun-Yan Zhu, Philipp Krahenbuhl and Alexei Efros. Generative Visual Manipulation on the Natural Image Manifold. European Conference on Computer Vision (ECCV)., pages 597–613, 2016. 3

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. Advances In Neural Information Processing Systems, pages 1–9, 2012. 1

[18] Chung-Wei Lee, Wei Fang, Chih-Kuan Yeh, and Yu-Chiang Frank Wang. Multi-Label Zero-Shot Learning with Structured Knowledge Graphs. CVPR, 2018. 3

[19] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-SGD: Learning to Learn Quickly for Few-Shot Learning. arXiv:1707.09835, 2017. 2

[20] Tsung Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 8693 LNCS, pages 740–755, 2014. 1, 5

[21] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. Proceedings of the IEEE International Conference on Computer Vision, 2015 Inter:3730–3738, 2015. 5, 8

[22] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. Least Squares Generative Adversarial Networks. IEEE International Conference on Computer Vision (ICCV), pages 1–16, 2016. 3

[23] Tsendsuren Munkhdalai and Hong Yu. Meta Networks. arXiv:1703.00837, 2017. 2

[24] Tushar Nagarajan and Kristen Grauman. Attributes as operators: factorizing unseen attribute-object compositions. Proceedings of the European Conference on Computer Vision (ECCV), 2018. 3

[25] Dennis Park and Deva Ramanan. Articulated pose estimation with tiny synthetic videos. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015-Octob:58–66, 2015. 1, 3

[26] Adam Paszke, Gregory Chanan, Zeming Lin, Sam Gross, Edward Yang, Luca Antiga, and Zachary Devito. Automatic differentiation in PyTorch. 31st Conference on Neural Information Processing Systems, (Nips):1–4, 2017. 4

[27] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv:1511.06434, pages 1–16, 2015. 3, 7

[28] Alexander J. Ratner, Henry R. Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to Compose Domain-Specific Transformations for Data Augmentation. (Nips), 2017. 3

[29] Sachin Ravi and Hugo Larochelle. Optimization As a Model for Few-Shot Learning. International Conference on Learning Representations (ICLR), pages 1–11, 2017. 2

[30] Scott Reed, Yutian Chen, Thomas Paine, Aaron van den Oord, S. M. Ali Eslami, Danilo Rezende, Oriol Vinyals, and Nando de Freitas. Few-shot autoregressive density estimation: towards learning to learn distributions. arXiv:1710.10304, (2016):1–11, 2018. 3

[31] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. IJCV, 9 2015. 1

[32] Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Abhishek Kumar, Rogerio Feris, Raja Giryes, and Alex M Bronstein. -Encoder: an Effective Sample Synthesis Method for Few-Shot Object Recognition. NIPS, 2018. 1, 3

[33] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical Networks for Few-shot Learning. Advances In Neural Information Processing Systems (NIPS), 2017. 2

[34] Hao Su, Charles R Qi, Yangyan Li, and Leonidas J Guibas. Render for CNN Viewpoint Estimation in Images Using CNNs Trained with Rendered 3D Model Views.pdf. IEEE International Conference on Computer Vision (ICCV), pages 2686–2694, 2015. 3

[35] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. arXiv:1512.00567, 2015. 3

[36] Antonio Torralba and Alexei A Efros. Unbiased Look at Dataset Bias. CVPR, pages 1521–1528, 2011. 8

[37] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching Networks for One Shot Learning. Advances In Neural Information Processing Systems (NIPS), 2016. 1, 2

[38] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. CNN-RNN: A Unified Framework for Multi-label Image Classification. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2285–2294, 2016. 3

[39] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-Shot Learning from Imaginary Data. arXiv:1801.05401, 2018. 3

[40] Mark Yatskar, Vicente Ordonez, Luke Zettlemoyer, and Ali Farhadi. Commonly Uncommon: Semantic Sparsity in Situation Recognition. CVPR, 2017. 3

[41] Aron Yu and Kristen Grauman. Semantic Jitter: Dense Supervision for Visual Comparisons via Synthetic Images. Proceedings of the IEEE International Conference on Computer Vision, 2017-Octob:5571–5580, 2017. 3

[42] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond Empirical Risk Minimization. arXiv:1710.09412, pages 1–11, 2017. 7

[43] Fengwei Zhou, Bin Wu, and Zhenguo Li. Deep Meta-Learning: Learning to Learn in the Concept Space. arXiv:1802.03596, 2 2018. 2

[44] Jun Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. Proceedings of the IEEE International Conference on Computer Vision, 2017-Octob:2242–2251, 2017. 1, 3