Learning Image Representations by Completing Damaged Jigsaw Puzzles

Dahun Kim
KAIST
mcahny@kaist.ac.kr

Donghyeon Cho
KAIST
cdh12242@gmail.com

Donggeun Yoo
KAIST
dgyoo@rcv.kaist.ac.kr

In So Kweon
KAIST
iskweon@kaist.ac.kr

Abstract

In this paper, we explore methods of complicating self-supervised tasks for representation learning. That is, we do severe damage to data and encourage a network to recover them. First, we complicate each of three powerful self-supervised task candidates: jigsaw puzzle, inpainting, and colorization. In addition, we introduce a novel complicated self-supervised task called "Completing damaged jigsaw puzzles" which is puzzles with one piece missing and the other pieces without color. We train a convolutional neural network not only to solve the puzzles, but also generate the missing content and colorize the puzzles. The recovery of the aforementioned damage pushes the network to obtain robust and general-purpose representations. We demonstrate that complicating the self-supervised tasks improves their original versions and that our final task learns more robust and transferable representations compared to the previous methods, as well as the simple combination of our candidate tasks. Our approach achieves state-of-the-art performance in transfer learning on PASCAL classification and semantic segmentation.

1. Introduction

The goal of representation learning is to learn robust and general-purpose visual features. Typically, the amount of labeled data decreases as the extent of annotation increases. The networks trained on limited amount of labeled data are easily overfitted and have poor representation ability. Representation learning is used to avoid this problem by pre-training visual features on large-scale data before training on target tasks.

Conventional yet still popular method to learn such features is to pre-train image classification [11, 20, 33, 34] on millions of human-labeled data such as ImageNet [32]. It provides powerful representations and image priors when the target task and data are similar. However, the dependency on human supervision of this traditional method limits its scalability and adaptability to dissimilar target tasks and domains(*e.g.* depth prediction).

(a) (b)

Figure 1. **Learning image representations by completing damaged jigsaw puzzles.** We sample 3-by-3 patches from an image and create damaged jigsaw puzzles. (a) is the puzzles after shuffling the patches, removing one patch, and decolorizing. We push a network to recover the original arrangement, the missing patch, and the color of the puzzles. (b) shows the outputs; while the pixel-level predictions are in *ab* channels, we visualize with their original L channels for the benefit of the reader.

Many researches have been conducted to minimize human supervision in computer vision. For example, weakly-supervised learning [10, 15–17, 27] has been proposed to learn object localization using *weak* image-level annotations rather than bounding boxes or pixel-level annotations. In the same vein, recent representation learning has also been improved to minimize human supervision. The emerging family of such methods is self-supervised learning; It manufactures a supervised task and labels from raw images, so that unlimited amount of labeled data can be used. A considerable number of such methods [4–6, 21, 22, 25, 26, 30, 36–39] have been proposed in last few years. They often train a network to infer geometrical configuration [4, 25], recover missing pixels [30] or channels [21, 38, 39] of images. The features learned by these methods have been successfully transferred to different target tasks, such as classification, detection, and semantic segmentation, and resulted in promising performances.

The common intuition of these approaches is that a network obtains useful representations of scenes and objects while struggling to solve a challenge task that requires high-level reasoning. Based on this idea, we propose a concept

of complicating a self-supervised task where we raise the difficulty of the task. More specifically, we design more difficult versions of jigsaw puzzle, inpainting, and colorization tasks. We investigate the effectiveness of our approach by transferring the learned features on PASCAL VOC classification, detection, and segmentation tasks [7, 8]. In order to further the idea, we design a task called "Completing damaged jigsaw puzzle", which is puzzles with one piece missing and the other pieces without color. Then, jigsaw puzzle, inpainting, and colorization tasks are jointly optimized. The network learned in this way preserves better feature representations for classification, detection and semantic segmentation.

In summary, our main contributions are as follows:

- We propose an approach of making self-supervised tasks more challenging for representation learning.
- We design a problem of completing damaged jigsaw puzzles where three different self-supervised tasks are complicated and incorporated simultaneously.
- We show that the representations learned by our approach achieve state-of-the-art performances on PASCAL classification and semantic segmentation [7, 8] when transferred on AlexNet, compared to existing self-supervised learning methods.

2. Related works

A considerable number of unsupervised learning approaches have been studied to learn image representations without relying on human-annotation. The most fundamental example is the autoencoder [35], which is a generative model that reconstructs the input data, aiming to extract the data representation. Since then, various generative models rooted in the autoencoder have been proposed. For example, DCGAN [31] and variational auto-encoders [3] have been proposed for further photorealistic reconstruction and feature learning.

Our study falls into *self-supervised learning* which has emerged as a new stream of unsupervised learning. This technique manufactures supervision signal from the raw visual data and achieves promising results in learning discriminative features. Recent methods commonly use images [4, 6, 21, 25, 26, 30, 38, 39], and often video [12, 24, 29, 36], or other sensory data such as egomotion and sound [1, 2, 13, 28].

Different supervision signals encourage the network to pay attention to different characteristics in images. Thus, the virtues of the learned representations also differ across the self-supervised tasks. Recent methods on self-supervised feature learning can be broadly categorized according to the type of knowledge preferred in the training: spatial configuration, context, and cross-channel relations.

Spatial Configuration. The methods that operate on the spatial dimension of images usually extract the patches from the image and learn the network to infer spatial relations between them. Doersch *et al*. [4] proposed a problem with 3-by-3 puzzles, where the network sees one of the outer patches, and predicts its relative position to the center patch. Noroozi and Favaro [25] learn image representations by solving the jigsaw puzzle with the 3-by-3 patches which imposes a challenging task of estimating what permutation has been used in shuffling. The learned features well capture the geometrical configuration of the objects as mentioned in [4].

Image Context. A contextual autoencoder was proposed by Pathak *et al*. [30] in order to drive representation learning. The supervisory signal comes from inpainting task where the network is encouraged to recover dropped part of the image from the surrounding pixels. Also, Isola *et al*. [12] exploited a co-occurance cues as a self-supervision where the network takes two isolated patches and predict whether or not they were taken from nearby locations in an image. These methods allow the network to learn contextual relations between part of an image and the rest or between each object parts/instances in an image.

Cross-Channel Relations. The methods that manipulate the images in channel domain have also been proposed. Typically, they remove one subset of the image channels, and train the network to recover it from the remaining channel(s). Zhang *et al*. [38] and Larsson *et al*. [21] obtain self-supervision from the task of colorization where the network predicts *ab* channels given L channel. Zhang *et al*. [39] took a one step further by learning colorization together with the inverse mapping from *ab* channels to L channel.

Combining Multiple Self-supervised Tasks. The aforementioned methods are essentially relying on a single supervisory signal. Recently, representation learning by multiple supervisory signals has also emerged. Zhang *et al*. [39] proposed a bidirectional cross-channel prediction to aggregate complementary image representations. They propose a network split into to two groups, and each subnetworks are trained separately. Wang *et al*. [37] exploited two self-supervised approaches to unify different types of invariance appearing in the two approaches. Doersch and Zisserman [5] combine multiple self-supervised tasks to create a single universal representation. However, each of the methods have limitations. In [39], splitting the network reduces the number of parameters by half which might limit the feature transferability. Also, [37] trains two tasks in sequential order. That is, the training on ranking video frames [36] comes only after the training on estimating relative position [4] finishes. Lastly, the involved tasks in [5] operate

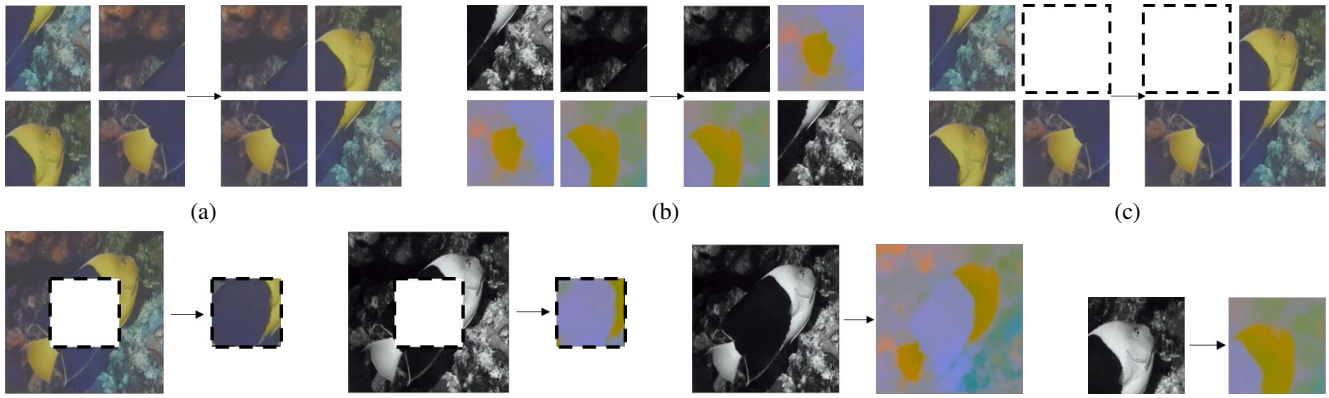

Figure 2. **Illustrations of complicating self-supervised tasks.** (a) Conventional 2×2 jigsaw puzzles. (b) Complicated 2×2 jigsaw puzzles; each patch's L or *ab* channel is dropped. (c) Complicated 2×2 jigsaw puzzles; one of the patches is completely dropped. (d) Conventional inpainting. (e) Complicated inpainting; it outputs in *ab* channels from an input in only L channel. (f) Conventional colorization. (g) Complicated colorization; only one-quarter of the entire image is given for colorization.

on very different inputs, which hinders simultaneous training of all tasks and requires special handlings.

Our study shares the goal with [37,39] and [5] where we want to learn representations that have all-round capability in every downstream task. However, our approach differs in the strategy; We squeeze a network to solve more complicated tasks, and in the same vain, our final method combines the complicated tasks and trains them simultaneously.

3. Approach

A number of recent self-supervised learning methods commonly operate via *damage-and-recover* mechanisms. In other words, the networks are supervised to recover intentionally damaged image data. For the purpose of representation learning, the damages are designed so that the recovery requires the high-level understanding of the objects and the scene. During training, the representations that are necessary for the recovery are learned, resulting in task/damage-specific features. For example, the spatial configuration is damaged in jigsaw puzzles, so the learned representations are focused on the configuration and geometry of objects. Similarly, the representations learned from inpainting and colorization preferably encode contextual and cross-channel relations as analyzed in [30] and [21], respectively.

Motivated by the mechanism above, we design a strategy where we drive the network to recover even more severe damage. More specifically, we do further damage to the data in jigsaw puzzle, inpainting, and colorization to make them more challenging as illustrated in Fig. 2. The methods of complicating each of the tasks are explained in Sec. 3.1. Furthermore, in order to maximize the effectiveness of our approach, we incorporate those three tasks in a single problem, "Completing damaged jigsaw puzzles", as detailed in Sec. 3.2.

3.1. Complicating Each Self-supervised tasks

In this section, we briefly review each of jigsaw puzzle, inpainting and colorization, and explain the methods of complicating them. Considering that different damages teach different lessons, we do additional damage to the data domains that have remained intact in the original task. The effectiveness of the complicated versions is quantitatively evaluated in Sec. 5.1.

Jigsaw Puzzle. With 2-by-2 puzzles, let us define S a sequence of puzzle patches X_1-X_4 shuffled by a permutation P. The spatial configuration of objects is intermixed by the permutation.

Accordingly, we consider two additional types of damage that make jigsaw puzzles more difficult. First, we do damage in the channel-wise domain, where half of the puzzles have only the L channel and the other half, *ab* channels, as shown in Fig. 2-(b). Successfully solving the puzzles requires not only the knowledge on spatial configuration, but also the understanding of the cross-channel relations between L and *ab* channels. Second, we damage the image context by removing one piece from a complete set of puzzles, as shown in Fig. 2-(c). In practice, a piece is discarded with a probability of 0.4 and the missing contents are replaced with Gaussian noise. Doing well on this task may require extra understanding on the full context without seeing the missing area.

As in [25], we train an AlexNet-based network to learn a mapping $\hat{P} = f_{\text{jig}}(S)$ to a probability distribution over 24(that is, 4!) possible permutations $\hat{P} \in [0,1]^{24}$ with loss,

$$\mathcal{L}_{jig} = -\sum P \log(\hat{P}). \quad (1)$$

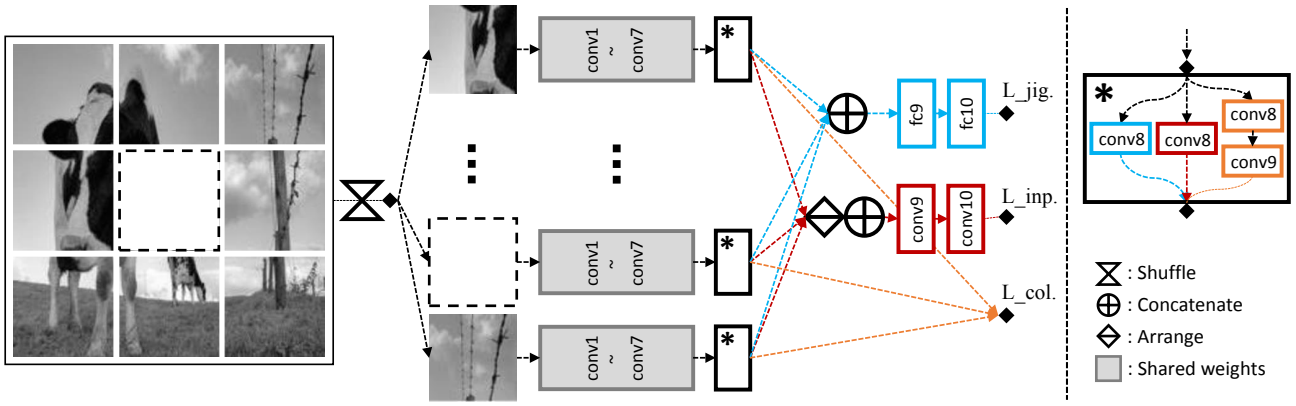

Figure 3. **The architecture for "Completing damaged jigsaw puzzles".** It is a 9-tower siamese network. The shared tower(colored in gray) consists of AlexNet $conv1$-7 layers. note $fc6$-7 are converted into equivalent $conv6$-7 layers for the pixel-level outputs. The task branches for jigsaw puzzle, inpainting, and colorization are marked in blue, red, and orange, respectively. The learned shared tower is used for transfer learning on downstream tasks.

Inpainting. Inpainting is a problem of restoring lost regions of an image. In the field of representation learning, a small patch X_p is removed from the image X, and remaining parts X_r are used for inferring the removed patch X_p. It is formulated as

$$\hat{X}_p = f_{\text{inp}}(X_r). \tag{2}$$

By solving this problem, the network learns contextual information of X_r and between X_r and X_p.

In order to do damage of a different flavor, we discard a subset of image channels. Unlike the original inpainting where all channels are given (Fig. 2-(d)), our complicated inpainting requires generation of *ab* channels of the missing region from the surrounding pixels in L channel (Fig. 2-(e)). While struggling to solve this problem, the network learns cross-channel relations as well as the contextual information. We use Euclidean distance between the prediction and the ground truth as a loss as proposed in [30] as

$$\mathcal{L}_{inp} = \left\| \hat{X}_p^{ab} - f_{\text{inp}}(X_r^L) \right\|_2^2, \tag{3}$$

where superscripts L and *ab* denote the input's L and *ab* channels, respectively.

Colorization. Colorization and other cross-channel prediction tasks [21, 38, 39] discard and recover a subset of image channels to learn cross-channel relations.

Additional damage for more difficult colorization takes place in the context domain. We encourage the network to see only part of images, and colorize in the absence of the full context. Specifically, we feed the network with the L channel of only one patch out of the 2×2 puzzles, and push it to colorize the patch as shown in Fig. 3.1-(g). The colorization becomes more difficult since only one-quarter of the entire image is available.

As in [38], the network learns a mapping $\hat{X^{ab}} = f_{col}(X^L)$ to a probability distribution over possible colors $\hat{X^{ab}} \in [0,1]^{313}$, where the *ab* are quantized to 313 values. We train the network with the 313-way classification loss as,

$$\mathcal{L}_{col} = -\sum v(X^{ab} \log(\hat{X^{ab}})), \tag{4}$$

where v(·) denotes a color-class rebalancing term.

3.2. Completing Damaged Jigsaw Puzzles

In order to further develop our idea, we design our final problem, "Completing damaged jigsaw puzzles", by involving all the damages and recoveries mentioned above. As its name indicates, this problem requires the simultaneous recovery of the following damages: (1) shuffling the image patches, (2) discarding one patch, (3) dropping *ab* channels in all the patches. During training, the network is encouraged to arrange the puzzles, recover the missing context, and colorize the patches. In practice, recovering the missing patch is defined as generating *ab* channels of the missing region from the surrounding pixels in L channel.

Recent self-supervised learning methods that use multiple self-supervised tasks either assign separate features to each tasks [39], train each tasks in sequential order [37], or jointly train the tasks [5]. We share with them the goal of learning a single set of well-rounded representations. However, our approach complicates each involved tasks to fuel the *damage-and-recover*, whereas the previous methods adopt the original form of existing self-supervised tasks. More specifically, our final problem involves a jigsaw puzzle with one piece missing, inpainting across channels, and

colorization with a narrower view, which are more complicated than their predecessors. Also, each tasks are intermingled in a way that some tasks share the knowledge. That is, the understanding of cross-channel relation supports both the colorization and the inpainting, and the contextual information is shared across all tasks. As a result, the network learns to effectively integrate and propagate the different knowledge on the spatial configuration, image context, and cross-channel relations into the final representations. Finally, all our involved tasks share the input space: a set of damaged puzzles. This makes our approach immune to the risk in [5, 39] that use different inputs for each tasks, where the network might task-specifically encode the representations depending on the type of inputs, as stated in [5].

In practice, our method operate on 3×3 puzzles rather than 2×2, for more discriminative representations.

Architecture and Losses. Our architecture is shown in Fig. 3. It is a 9-tower siamese network as in [25]. The shared tower follows the standard AlexNet [20] to provide a fair comparison with recent self-supervised learning methods [4, 6, 21, 25, 26, 30, 36–39]. The task branches of the jigsaw puzzle, inpainting, and colorization are rooted to the shared tower, and colored in blue, red, and orange, respectively.

In the jigsaw branch, 9 sets of the common features ($conv7$ features) pass through a fully-connected layer, $fc8$(blue), and are concatenated, then fed into two more fully-connected layers up to $fc10$(blue), resulting in a 1000-long vector. We use the same $\mathcal{L}_{jig}$ as Eq. (1). In the inpainting branch, the 9 features go through a 1×1 convolutional layer. This time, we arrange the features before concatenating them as we know what permutation has been used in the inputs. After two more 1×1 convolutions($conv9$, $conv10$, red), the features have a volume of $7 \times 7 \times 313$, where 313 denotes the number of quantized color values as in [38]. Note that we use a classification loss rather than Eq. (3) as,

$$\mathcal{L}_{inp}^{cls} = -\sum v(X_p^{ab} \log(\hat{X_p^{ab}})), \tag{5}$$

where $\hat{X_p^{ab}}$ denotes the predicted chromaticity values of the missing puzzle. Each of the 9 features is fed into the colorization branch, resulting in 9 branches. Each branch is an equivalent form of the network in [38] which has two more 1×1 convolutions($conv8$, $conv9$, orange) after the shared tower, resulting in features of $7 \times 7 \times 313$. Our colorization loss is a sum of the 9 losses of Eq. (7) as,

$$\mathcal{L}_{col} = -\sum_{i=1}^{9}(\sum v(X_i^{ab} \log(\hat{X_i^{ab}}))), \tag{6}$$

where X_i denotes i*th* of the input patches. Finally, our loss for "Completing damaged jigsaw puzzles" is the sum of the three losses as,

$$\mathcal{L}_{final} = \mathcal{L}_{jig} + \alpha\mathcal{L}_{inp}^{cls} + \beta\mathcal{L}_{col}, \tag{7}$$

where α and β are weighting parameters.

Simple Combination. We also consider combining the original forms of self-supervised tasks, conceptually following [5]. We jointly train original versions of the three tasks: jigsaw puzzles, inpainting, and colorization. Although the types of involved tasks are different to [5], we provide a self-comparison on the effectiveness of our approach and the simple combination in Sec. 5.3.

4. Training

We train our proposed network on 1.3M images from the training set of ImageNet without annotations. We resize the input images to 312×312 pixels, and extracted patches of 140×140 and 85×85, in 2-by-2 and 3-by-3 puzzles, respectively. We use caffe [14] for implementation. The network is trained by ADAM optimizer [18] for 350K iterations with batch size of 64 on a machine with a GTX 1080-Ti GPU and an intel i7 3.4GHz CPU. The learning rate is set to 10^{-3}, and is dropped by a factor of 0.1 every 100K iterations. We use α, β = 0.01 for the experiment in Sec. 3.2. Inpainting and colorization of Sec. 3.1 follow the protocol of their original papers [30, 38], respectively.

5. Results and Discussions

In this section, we provide both quantitative and qualitative evaluations and discussions of our self-supervised learning approach. Further transfer learning results on new tasks(*e.g.* depth prediction) and with deeper network(*e.g.* vgg [33]) are presented in our supplementary material.

5.1. Fine-tuning on PASCAL

In this section we evaluate the effectiveness of both the "Complicating each self-supervised tasks" in Sec. 3.1 and our final task, "Completing damaged jigsaw puzzles" in Sec. 3.2. To do this, we transfer the learned representations to a standard AlexNet [20] and rescale the weights via [19]. We test on some or all of the PASCAL tasks, using VOC 2007 [7] for classification and detection, VOC 2012 [8] for segmentation; these are standard benchmarks for representation learning.

5.1.1 Complicating each self-supervised task

In Sec. 3.1, we explore the idea of complicating the jigsaw puzzle, inpainting, and colorization to benefit representation learning. We evaluate the effectiveness of each complications by comparing the performances before and after

Method	Complication	Class.	Segm.
Jigsaw(Sec. 3.1)	None	64.7	34.9
Jigsaw(Sec. 3.1)	L-or-ab dropped	65.5	35.7
Jigsaw(Sec. 3.1)	A piece removed	65.3	35.7
Inpainting [30]	None	56.5	29.7
Inpainting(Sec. 3.1)	Cross-Channel	57.7	30.2
Colorization [38]	None	65.9	35.7
Colorization(Sec. 3.1)	Narrow view	66.7	36.8

Table 1. **Effectiveness of complicating self-supervised tasks on PASCAL.** Classification is evaluated on PASCAL VOC 2007 with testing frameworks from [19], using mean average precision(mAP) as a performance measure. Segmentation is evaluated on PASCAL VOC 2012 with testing framework from [23], which reports mean intersection over union(mIU).

the complications in downstream tasks: classification and semantic segmentation.

The results are shown in Table. 1. In all cases, the complicated self-supervised tasks consistently achieve higher scores than their predecessors both in classification and segmentation. These results indicate that the capacity of the network was still above the difficulty of the existing self-supervised tasks, and that indeed, useful representations can be extracted more via solving more difficult tasks.

5.1.2 Completing Damaged Jigsaw Puzzles

We evaluate how beneficial is our final self-supervised task, "Completing damaged jigsaw puzzles", in learning representations. We transfer the learned weights from the shared tower Fig. 3 on classification, detection, and semantic segmentation. As shown in Table. 2, our method outperforms all the previous methods in classification and segmentation, and achieves the second best performance in the detection task, even though the network has been exposed only on grayscale images during pretraining. We also summarize the comparison on classification and segmentation tasks in Fig. 4 which indicates that our approach learns more robust and general-purpose representations in comparison to each of the involved tasks and all the conventional methods.

5.2. Linear Classification on ImageNet

We test the task-generality of our learned representations on large-scale representation learning benchmarks. As proposed in [38], we freeze each layer of our learned features from $conv1$ to $conv5$, and initialize the subsequent unfrozen layers with random values. Then, we train linear classifiers on top of each layer on labeled ImageNet [32] dataset.

The result is shown in Table. 3. ImageNet-pretrained AlexNet shows the best performance and is the upper bound in this comparison. Since our network only learns from L channel, $conv1$ features suffer lack of input information, resulting in slightly lower score compared to other methods. However, it overcomes this handicap immediately from $conv2$ layer, and achieves competitive performances in higher layers. Finally, $conv4$ and $conv5$ features achieve the second best and state-of-the-art performances, respectively.

As shown in [25], the last layers of the pretrained network tend to be task-specific, while the first layers are general-purpose. In our proposed architecture(Fig. 3), this transition from general-purpose to task-specific is delayed and left to the task branches. Since the last features of the shared tower must support all three different, they should remain as general as possible, rather than get biased to either of the tasks. Also, the network can hardly assign separate features to each tasks since the features required by the tasks often overlap, thus it has to integrate and hold the different features up to the last layers.

Method	Class.	Det.	Segm.
ImageNet [20]	79.9	56.8	48.0
Random	53.3	43.4	19.8
RelativePosition [4]	65.3	51.1	-
Jigsaw [25]	67.6	**53.2**	37.6
Ego-motion [36]	54.2	43.9	-
Adversarial [6]	58.6	46.2	34.9
Inpainting [30]	56.5	44.5	29.7
Colorization [38]	65.9	46.9	35.6
Split-Brain [39]	67.1	46.7	36.0
ColorProxy [21]	65.9	-	38.4
WatchingObjectMove [29]	61.0	52.2	-
Counting [26]	67.7	51.4	36.6
CDJP	**69.2**	52.4	**39.3**

Table 2. **Evaluation of transfer learning on PASCAL.** Classification and detection are evaluated on PASCAL VOC 2007 with testing frameworks from [23] and [9], respectively. Both tasks are evaluated using mean average precision(mAP) as a performance measure. Segmentation is evaluated on PASCAL VOC 2012 with testing framework from [23], which reports mean intersection over union(mIU).

Method	conv1	conv2	conv3	conv4	conv5
ImageNet [20]	19.3	36.3	44.2	48.3	50.5
Random	11.6	17.1	16.9	16.3	14.1
RelativePosition [4]	16.2	23.3	30.2	31.7	29.6
Jigsaw [25]	**18.2**	28.8	34.0	33.9	27.1
Adversarial [6]	14.1	20.7	21.0	19.8	15.5
Inpainting [30]	17.7	24.5	31.0	29.9	28.0
Colorization [38]	12.5	24.5	30.4	31.5	30.3
Split-Brain [39]	17.7	29.3	**35.4**	**35.2**	32.8
Counting [26]	18.0	**30.6**	34.3	32.5	25.7
CDJP	14.5	27.2	32.8	34.3	**32.9**

Table 3. **Linear classification on ImageNet.** We train linear classifiers on top of each layer of the learned feature representations. We use publicly available testing code from [38] and report top-1 accuracy of AlexNet on ImageNet 1000-way classification. The learned weights between $conv1$ and the displayed layer are frozen.

Combination	Class.	Segm.
Jig.	66.6	36.8
Jig.+Inp.	67.4	37.9
Jig.+Col.	68.4	38.6
Jig.+Inp.+Col./simple	68.0	38.1
Jig.+Inp.+Col.(CDJP)	**69.2**	**39.3**

Table 4. **Comparing different combinations of self-supervised tasks on PASCAL.** we evaluate different combinations of self-supervised tasks on PASCAL classification and segmentation in the same setting as in Table. 1. We make different combinations using our architecture (Fig. 3) with or without certain task branches; this may have caused slight performance differences from the original task. We also report the result of simple combinations where the original versions of each tasks are jointly trained.

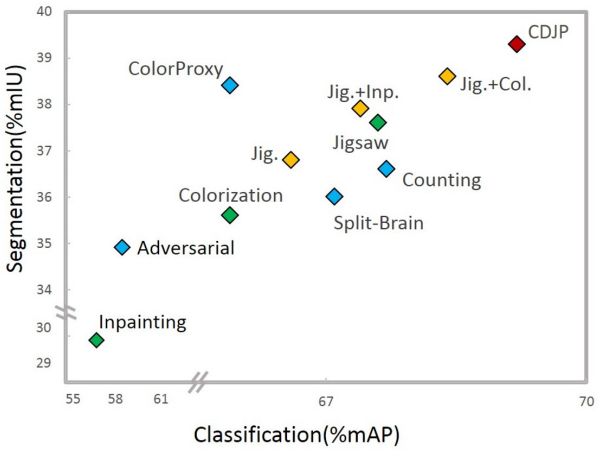

Figure 4. **Summarization of performances of different self-supervised learning methods and combinations.** We compare the state-of-the-art methods(Table. 2), our final method (CDJP), and each involved tasks in our final method and the simple combination(Table. 4). The involved tasks, their original versions, the simple combination, the other existing methods, and our final method are marked in orange, green, gray, blue and red, respectively. Note that $Jig.$ is what we reproduced in our architecture.

5.3. Comparing Combinations of Self-supervised tasks

In order to show the impact of each task, we evaluate different combinations on PASCAL classification and semantic segmentation tasks. We experiment with the same architecture Fig. 3, but with or without certain task branches to make different combinations. In addition, as we mentioned in Sec. 3.2, we provide the result of the simple combination of the tasks in their original form, which conceptually follows [5].

The results are shown in Table. 4. We set the jigsaw puzzle as our starting point and add different tasks to it. We can see that the performances increase every time the tasks are combined. Our final method which combines all three tasks obtains the best scores and improves our jigsaw puzzle by 2.6% and 2.5%scores both in classification and semantic segmentation tasks. The simple combination of the original versions slightly improves their single-task baselines [25, 30, 38] in both test tasks, but not better than our $Jig.$+$Col.$ and $Jig.$+$Inp.$+$Col.$(CDJP) methods.

5.4. Nearest Neighbor Search

The pretrained networks recognize the semantic similarity of data by their own standards. We qualitatively evaluate the validity of this reasoning of the networks by performing 'nearest neighbor search' which has been proposed in [4] and further used in [26, 37]. In this experiment we compare AlexNets [20] pretrained by different methods: jigsaw puzzle [25], inpainting [30], colorization [38], ours, and ImageNet classification [20]. We perform retrieval on $fc6$ (the feature before the concatenation) for jigsaw puzzle, $conv5$ (the last layer of the encoder) for inpainting, and $conv7$/$fc7$ features for the remaining methods.

Single-task Baselines. As in figure 5, the learned representations in each methods show distinct characteristics. For example, the jigsaw puzzle representations retrieve objects with the same pose and shape. Even in the *blurred* image, it retrieves objects with similar silhouettes. In inpainting, objects that would co-occur or share the similar background are retrieved, such as things to ride for *horse* and caregivers for *baby*. The features learned by colorization is often color-specific, and retrieves babies wearing pink clothes for *baby*, and sometimes false samples with blue-green color for *bottle*. Also, blurred objects a retrieved for the *blurred* image. Such color-sensitivity sometimes misrepresent semantics, *e.g.* a brown chair back is retrieved for *horse* image.

Similarity to ImageNet Classification Pretraining. Note that we consider pretraining on ImageNet classification as our gold standard in this qualitative evaluation. Our approach integrates the characteristics of the single-task baselines, yet mostly complement and overcome the aforementioned sensitivities. First, our approach is more invariant to pose/viewpoint variations compared to jigsaw puzzle baseline, and represents *horse*s and *baby*s in different pose and viewpoint as semantically nearby, which is also the case in 'ImageNet' model. Furthermore, our representations are more robust in intra-class color variations, and retrieves objects with various colors according to *horse*, *baby*, and *bottle* query images, which also raise our model closer to our gold standard. Our model also adopts the virtues of the single-task baselines. To illustrate, for *blurred* object, as in colorization, our model retrieves images that are semantically ambiguous. We can see the same tendency in the

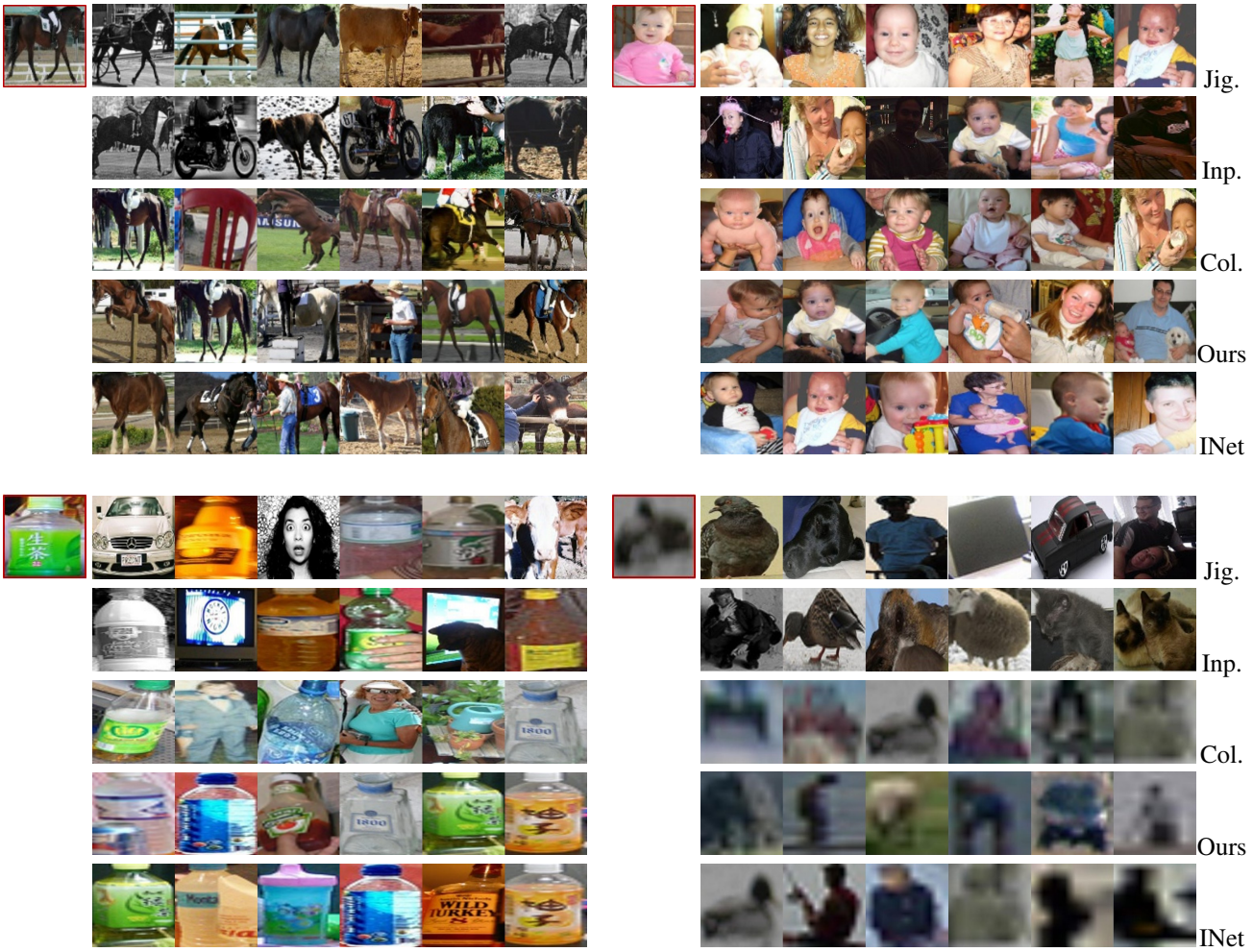

Figure 5. **Nearest Neighbor Search.** We perform image retrieval on the object instances cropped from the PASCAL VOC 2012 [8] *trainval* dataset. The query images are in red boxes. Down from the top rows are the retrieval results of jigsaw puzzle, inpainting, colorization, our method, and ImageNet classification, respectively.

'ImageNet' model, where it may consider the query image to be vague, and retrieves also blurred objects in different categories. Finally, our model adopts a reasonable understanding on the image context, which enabled the retrieval of co-occurable objects, *e.g.,* person with *horse* and parent with *baby*. Interestingly, we observe that the 'ImageNet' retrieves images where person and horse; caregiver and baby appear together, similarly to ours. These results can be viewed as one reason that our approach can propagate the high-level semantics through our model, and raise its robustness and task generality of our representations.

6. Conclusions

In this paper, we study complicating self-supervised tasks for representation learning. We propose complicated versions of jigsaw puzzles, inpainting and colorization and show their effectiveness on representation learning. Furthermore, we design "Completing damaged jigsaw puzzles" as a more complicated and complex problem for self-supervised representation learning. While learning to recover and colorize original image content simultaneously, rich and general-purpose visual features are encoded into the network. Experiments contain transfer learning on PASCAL VOC classification, detection and segmentation, ImageNet linear classification as well as nearest neighbor search. All of the results clearly show that the features learned by our method generalize well across different high-level visual tasks.

Acknowledgements This research is supported by the Study on representation learning for object recognition funded by the Samsung Electronics Co., Ltd (Samsung Research)

References

[1] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, 2015.

[2] R. Arandjelovic and A. Zisserman. Ambient sound provides supervision for visual learning. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, 2017.

[3] M. W. Diederik P Kingma. Auto-encoding variational bayes. In *Proc. of Int'l Conf. on Learning Representations (ICLR)*, 2014.

[4] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, 2015.

[5] C. Doersch and A. Zisserman. Multi-task self-supervised visual learning. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, 2017.

[6] J. Donahue, P. Krähenbühl, and T. Darrelln. Adversarial feature learning. In *Proc. of Int'l Conf. on Learning Representations (ICLR)*, 2017.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[9] R. Girshick. Fast r-cnn. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, 2015.

[10] H.Bilen and A. Vedaldi. Weakly supervised deep detection networks. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2016.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2016.

[12] P. Isola, D. Zoran, D. Krishnan, and E. H. Adelson. Learning visual groups from co-occurrences in space and time. In *ICLR Workshop*, 2015.

[13] D. Jayaraman and K. Grauman. Learning image representations tied to egomotion from unlabeled video. *Int'l Journal of Computer Vision (IJCV)*, 125(1-3):136–161, 2017.

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[15] Z. Jie, Y. Wei, X. Jin, J. Feng, and W. Liu. Deep self-taught learning for weakly supervised object localization. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2017.

[16] V. Kantorov, M. Oquab, M. Cho, and I. Laptev. Contextlocnet: Context-aware deep network models for weakly supervised localization. In *Proc. of European Conf. on Computer Vision (ECCV)*, 2016.

[17] D. Kim, D. Cho, and D. Yoo. Two-phase learning for weakly supervised object localization. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, 2017.

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. of Int'l Conf. on Learning Representations (ICLR)*, 2015.

[19] P. Krähenbühl, C. Doersch, J. Donahue, and T. Darrell. Data-dependent initializations of convolutional neural networks. In *Proc. of Int'l Conf. on Learning Representations (ICLR)*, 2016.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. of Neural Information Processing Systems (NIPS)*, 2012.

[21] G. Larsson, M. Maire, and G. Shakhnarovich. Colorization as a proxy task for visual understanding. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2017.

[22] H.-Y. Lee, J.-B. Huang, M. Singh, and M.-H. Yang. Unsupervised representation learning by sorting sequences. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, 2017.

[23] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2015.

[24] I. Misra, C. L. Zitnick, and M. Hebert. Shuffle and learn: Unsupervised learning using temporal order verification. In *Proc. of European Conf. on Computer Vision (ECCV)*, 2016.

[25] M. Noroozi and P. Favaro. Unsupervised visual representation learning by context prediction. In *Proc. of European Conf. on Computer Vision (ECCV)*, 2016.

[26] M. Noroozi, H. Pirsiavash, and P. Favaro. Representation learning by learning to count. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, 2017.

[27] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free? - weakly-supervised learning with convolutional neural networks. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2015.

[28] A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba. Look, listen and learn. In *Proc. of European Conf. on Computer Vision (ECCV)*, 2016.

[29] D. Pathak, R. B. Girshick, P. Dollár, T. Darrell, and B. Hariharan. Learning features by watching objects move. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2017.

[30] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2016.

[31] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proc. of Int'l Conf. on Learning Representations (ICLR)*, 2016.

[32] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *Int'l Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2015.

[35] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408, 2010.

[36] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, 2015.

[37] X. Wang, K. He, and A. Gupta. Transitive invariance for self-supervised visual representation learning. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, 2017.

[38] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *Proc. of European Conf. on Computer Vision (ECCV)*, 2016.

[39] R. Zhang, P. Isola, and A. A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2017.