OCEAN: Object-centric arranging network for self-supervised visual representations learning

Changjae Oh\textsuperscript{a}, Bumsub Ham\textsuperscript{a}, Hansung Kim\textsuperscript{b}, Adrian Hilton\textsuperscript{b}, Kwanghoon Sohn\textsuperscript{a, \ast}

\textsuperscript{a}The School of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, South Korea
\textsuperscript{b}Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, Guildford GU2 7XH, United Kingdom

\textbf{A R T I C L E  I N F O}

Article history:
Received 16 May 2018
Revised 21 January 2019
Accepted 29 January 2019
Available online 6 February 2019

Keywords:
Self-supervised learning
Visual representations learning
Object proposals
Convolutional neural networks

\textbf{A B S T R A C T}

Learning visual representations plays an important role in computer vision and machine learning applications. It facilitates a model to understand and perform high-level tasks intelligently. A common approach for learning visual representations is supervised one which requires a huge amount of human annotations to train the model. This paper presents a self-supervised approach which learns visual representations from input images without human annotations. We learn the correct arrangement of object proposals to represent an image using a convolutional neural network (CNN) without any manual annotations. We hypothesize that the network trained for solving this problem requires the embedding of semantic visual representations. Unlike existing approaches that use uniformly sampled patches, we relate object proposals that contain prominent objects and object parts. More specifically, we discover the representation that considers overlap, inclusion, and exclusion relationship of proposals as well as their relative position. This allows focusing on potential objects and parts rather than on clutter. We demonstrate that our model outperforms existing self-supervised learning methods and can be used as a generic feature extractor by applying it to object detection, classification, action recognition, image retrieval, and semantic matching tasks.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

Recently, convolutional neural networks (CNN) have been applied to a variety of tasks, including object and action recognition (Aghamaleki & Baharlou, 2018; Donahue et al., 2014; Gkioxari, Girshick, & Malik, 2015; Nwaeke, Teh, Al-garadi, & Alo, 2018; Simonyan & Zisserman, 2014), detection (Girshick, 2015; Girshick, Donahue, Darrell, & Malik, 2014; Ohn-Bar & Trivedi, 2017), semantic segmentation (Shelhamer, Long, & Darrell, 2017; Simonyan & Zisserman, 2014), tracking (Li, Li, & Porikli, 2016), and visual correspondence (Zbontar & LeCun, 2016), and have produced satisfactory results. They typically use supervised learning approaches that require a huge amount of annotated images (Donahue et al., 2014; Girshick, 2015; Girshick et al., 2014; Zbontar & LeCun, 2016), e.g., object-level bounding boxes for object detection. Contrary to the aforementioned approach, unsupervised learning does not require manual annotations for the purposes of learning abstract features (e.g., semantic objects) while showing low performance (Bengio, Courville, & Vincent, 2013; Hinton, 2007; Hinton & Salakhutdinov, 2006; Kingma & Welling, 2013).

A recurring theme in unsupervised learning is the use of self- (or meta-) supervision (Doersch, Gupta, & Efros, 2015; Gao, Jayaraman, & Grauman, 2016; Larsson, Maire, & Shakhnarovich, 2016; Misra, Zitnick, & Hebert, 2016; Pathak, Krahenbuhl, Donahue, Darrell, & Efros, 2016; Wang & Gupta, 2015; Zhang, Isola, & Efros, 2016). This refers to a network trained for a pretext (or proxy) task, which is not of direct interest, but significantly relates to the final high-level task, e.g., object detection, classification, and action recognition (Girshick, 2015; Gkioxari et al., 2015; Simonyan & Zisserman, 2014; Sun et al., 2017). Automatic image colorization (Larsson et al., 2016; Zhang et al., 2016) is a typical example of a pretext task; naturally coloring grey images requires prior knowledge of natural image appearance. Other pretext tasks include spatially arranging patches from a static image (Doersch et al., 2015; Noroozi & Favaro, 2016), reconstructing temporal ordering from shuffled video frames (Lee, Huang, Singh, & Yang, 2017; Misra et al., 2016), metric learning with object-like regions (Gao et al., 2016; Wang & Gupta, 2015), reconstructing different domain images (Zhang, Isola, & Efros, 2017), and image inpaint-
ing (Pathak et al., 2016). The main issue in self-supervised learning is designing the pretext task which is difficult to solve without an understanding of image semantics such as object and object parts.

In this paper, we propose an Object-Centric Arranging Network, called OCEAN, for self-supervised visual representations learning, that learns semantic features without any manual annotation. Motivated by Doersch et al. (2015) and Noroozi and Favaro (2016), we design a pretext task as rearranging patches from an input image. The key difference from previous work is the use of object proposals as arranging primitives instead of uniformly (or regularly) sampled patches (Fig. 1). Object proposal methods generate object-like regions (e.g., object parts) effectively at various scales with a high recall (Krähenbühl & Koltun, 2014; Manen, Guillaumin, & Van Gool, 2013; Uijlings, Van De Sande, Gevers, & Smeulders, 2013; Zitnick & Dollár, 2014). Instead of using uniformly sampled patches (Doersch et al., 2015; Noroozi & Favaro, 2016) that may include clutter (Fig. 1), the proposed rearranging task focuses on potential objects and object parts by employing generic object proposals. We demonstrate that with OCEAN we can achieve significantly higher performance.

The overview of our method is shown in Fig. 2. Experimental results show that our method yields competitive results compared to existing self-supervised learning approaches in various tasks including object and action detection, classification, image retrieval, and semantic correspondence.

Contributions: The major contributions of this paper are summarized as follows.

- We design a novel pretext task for self-supervised learning, according to which, geometric relationships (relative overlap, inclusion, and exclusion) between proposals are learnt effectively.
- We demonstrate the advantage of our model over other self-supervised learning methods by applying it to object detection, classification, and action recognition, which gives a mean average precision (mAP) of 48.6%, PASCAL VOC 2007 detection and classification dataset, and 50.3% in the PASCAL VOC 2012 action classification dataset, respectively.
- To verify that our model can be used as a generic feature extractor, we further apply it to other computer vision tasks including image retrieval and semantic correspondence.

2. Related works

Unsupervised learning encompasses a broad range of topics (Bengio et al., 2013). In the remainder of this section, we briefly describe representative works which are related to ours.

2.1. Unsupervised feature learning

Representative unsupervised learning methods include variational autoencoders (VAE) (Kingma, Mohamed, Rezende, & Welling, 2014; Kingma & Welling, 2013) and Boltzmann machines (Fischer & Igel, 2014; Hinton, 2007). These approaches attempt to learn generative models that capture the probabilistic distribution that is associated with the input latent variables of datasets. Recently, generative adversarial networks (GANs) (Goodfellow et al., 2014) have been introduced and, according to this approach, generative nets compete against discriminative ones to generate realistic high-resolution images. GANs can also be used for unsupervised visual representations learning by adding image encoders in a training stage (Donahue, Krähenbühl, & Darrell, 2016). However, unsupervised approaches in feature learning are still remaining challenging in terms of training neural networks for high-level tasks. They fail to encode the visual semantics for object detection and classification.

2.2. Self-supervised learning

Self-supervised and unsupervised learning are similar in that they do not require manual annotations. The difference between these two approaches is that self-supervised learning uses alternate forms of supervision that can be imposed algorithmically. These approaches try to solve pretext tasks where ground truths are freely available. Although the outputs of these tasks may not be of direct interest, an ability to answer pretext questions indicates that learned models implicitly embed semantic visual representations, and thus can be used to solve high-level vision problems such as object detection and classification. Doersch et al. design a novel pretext task with the goal of arranging shuffled pairs of patches from an input image to proper locations (Doersch et al.,
That is, the network estimates the relative location of a target patch from a reference patch. Noroozi and Favaro extend the idea of Doersch et al. (2015) by training a network to spatially arrange nine patches with random permutation, enabling the network to learn feature more effectively (Noroozi & Favaro, 2016). Pathak et al. propose a network for image inpainting that fills missing regions in an input image (Pathak et al., 2016). They assume that the network should learn semantic features to perform such a pretext task, meaning that the trained network can be used to extract generic features. Under a similar assumption, Zhang et al. (2016) and Larson et al. (2016) propose networks to colorize grayscale images. To generalize colorization as a domain transfer approach, Zhang et al. further predict the precision of one subset of the data channels from another (Zhang et al., 2017). Additionally, temporal relationships between frames can be employed to self-supervised learning. Jayaraman and Grauman present egomotion constraints from video to learning visual representations (Jayaraman & Grauman, 2015). Misra et al. propose the network to verify the temporal order of video sequences, and Lee et al. generalized this approach by extending problems by employing multiple-tuple of frames. Different from previous approaches that use video sequences, Pathak et al. learn to segment object which is provided by unsupervised motion segmentation from video (Pathak, Girshick, Dollár, Darrell, & Hariharan, 2017).

Our work is closely related to patch arrangement tasks (Doersch et al., 2015; Noroozi & Favaro, 2016). As illustrated in Fig. 1 (lower-left), previous methods use uniformly sampled patches that often contain distracting parts (e.g., background clutter and homogeneous regions) unless objects span the entire image. The uniformly sampled patches make it difficult for the network to learn semantic visual representations. Unlike these methods, we use object proposals that contain objects or parts as inputs to our arrangement task to learn a set of non-uniform patches which represent the object rather than background clutter. Semantic information is captured more effectively in this way by learning geometric relationships (including relative position, overlapping, inclusion, and exclusion) between object proposals as illustrated in Fig. 2.

3. Object-centric arranging network

3.1. Object proposals for visual representation

Object proposal methods extract several object candidates with high recall in a class-agnostic manner (Krähenbühl & Koltun, 2014; Manen et al., 2013; Uijlings et al., 2013; Zitnick & Dollár, 2014). In object detection (Cai, Tan, & Tan, 2017; Girshick, 2015; Girshick et al., 2014; He, Zhang, Ren, & Sun, 2015; Xu, Li, Wu, & Luo, 2017), the practice of using object proposals exhibits an advantage over the traditional sliding window approach: the search space and false alarms due to background clutter. This indicates that visual representations can be effectively captured by using object proposals as primitive units for further applications. Moreover, in this work we show that we can generate object proposals without any supervision. Thanks to these properties, object proposals have been used in various computer vision tasks, including visual tracking (Zhu, Porikli, & Li, 2016), action recognition (Gkioxari et al., 2015), and semantic correspondence (Ham, Cho, Schmid, & Ponce, 2016).

Recently, object proposals have been used to learn visual representations in video sequences (Gao et al., 2016). Nearby object proposals are sampled from consecutive frames. The network then performs metric learning, so that these proposals are closely embedded in feature space. Our method also uses object proposals for self-supervised learning in images or video, but solves a different pretext task within static images.

We assume that classifying geometric relationships between object proposals requires knowledge of semantic visual representations such as objects and their spatial layout, geometry, parts, and even categories. The framework of the proposed method and the network architectures are shown in Fig. 3 and Table 1, respectively.

![Fig. 3. Our framework for self-supervised visual learning. Feature vectors of input object proposals are extracted from convolutional layers followed by region-of-interest (ROI) pooling and fully-connected (FC) layers. We select a pair of FC6 outputs, and then concatenate (+) them. Through two fully-connected layers (FC7 and FC8) and a softmax layer, we classify the geometric relationship between a pair of proposals (relative position, overlap, inclusion, and exclusion). We train all convolutional and FC layers from scratch. More detailed network configuration is shown in Table 1.](image)

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer</td>
</tr>
<tr>
<td>conv1+relu + norm+pool</td>
</tr>
<tr>
<td>conv2+relu + norm+pool</td>
</tr>
<tr>
<td>conv3+relu</td>
</tr>
<tr>
<td>conv4+relu</td>
</tr>
<tr>
<td>conv5+relu</td>
</tr>
<tr>
<td>ROI pooling</td>
</tr>
<tr>
<td>FC6+relu</td>
</tr>
<tr>
<td>concat</td>
</tr>
<tr>
<td>FC7+relu</td>
</tr>
<tr>
<td>FC8</td>
</tr>
</tbody>
</table>
3.2. Network architecture

3.2.1. Convolutional and region-of-interest (ROI) pooling layers

We use a network similar to the AlexNet as our base feature extractor in order to ensure a fair comparison to other methods (Doersch et al., 2015; Pathak et al., 2016; Wang & Gupta, 2015). The convolutional layers can be replaced by other networks without loss of generality. We add an ROI pooling layer on top of the convolutional ones to extract features of object proposals that may have different sizes (Girshick, 2015). It generates the same size of feature vectors for all proposals. This ROI pooling layer enables the efficient computation of feature vectors, and this results in reduced training time.

3.2.2. Fully-connected layers and pairwise classification

Each output of the ROI pooling layer goes through the fully-connected layer, FC6. Two FC6 outputs, where each output is associated to two proposals, are concatenated subsequently. Given two proposals, we consider one as a reference proposal and the other as a target. The concatenated feature vectors are passed to the subsequent fully-connected layers (FC7 and FC8) and the following softmax layer in order to predict geometric relationships between two proposals. To this end, we consider this problem as a classification task. Given a reference proposal, we set the relative positions of a target proposal to the values top, left, right, bottom. We also consider overlap, reference inclusion, target inclusion, exclusion relationships between proposals, where reference (resp. target) inclusion indicates that the reference (resp. target) proposal spatially belongs to the target (resp. reference) proposal.

There are a few differences between our approach and existing methods (Doersch et al., 2015; Noroozi & Favaro, 2016). Note that existing methods use non-overlapping patches as inputs to prevent the CNN from simply learning discriminative low-level features. Our approach allows the overlapping and even inclusion cases as pretext tasks. Since OCEAN generates feature vectors of fixed size from inputs of various size by means of the ROI pooling architecture, the trivial shortcuts of Doersch et al. (2015) and Noroozi and Favaro (2016) can be alleviated. We show that the proposed method is free of the trivial shortcuts in Section 4.2, where it is demonstrated that high-level information is learned by the CNN within the pretext task.

3.3. Object proposal selection

Unsupervised learning from object proposals requires proposal selection due to the potentially large set of candidates as shown in Fig. 4(b). Simply choosing random object proposals may be insufficient to describe objects and their parts in an image, since many object proposals are generated from the background as shown in Fig. 4(c). We perform object proposal selection.

Object proposals are initially extracted by the SelectiveSearch (SS) algorithm (Uijlings et al., 2013). We then discard object proposals larger than 70% and smaller than 5% of the image area, which might distract the CNN from learning the object semantics. A greedy non-maximum suppression is then adopted to reduce the region candidates which are heavily overlapped, with the threshold of intersection-over-union (IoU) set to 0.4. Finally, we randomly select a pre-defined number of object proposals.

3.4. Implementation details

We use VLFeat MatConvNet toolbox (Vedaldi & Lenc, 2015), and train OCEAN with the ILSVRC2012 training set.

---

1 When nine object proposals are used as inputs, 9 × 8 = 72 pairs of proposal can be generated to train our network.

---

Fig. 4. Proposals selection. (a) An input image, (b) extracted object proposals, (c) randomly selected proposals, and (d) proposals used to training. Object proposals might be frequently generated from similar regions due to their characteristics of the repeatability. This might distract input proposals to imply object and their parts in the image.

which contains about 1.3M images with annotations (Deng et al., 2009). We use Adam to ensure an efficient stochastic optimization (Kingma & Ba, 2014). In the experimental trials, we use the same size of weights for layers conv1 to conv5 in AlexNet (Krizhevsky, Sutskever, & Hinton, 2012) and nine object proposals are randomly selected among the processed proposal set for every iteration. Experiments were performed with a 12GB NVIDIA Titan GPU and i7-5820K CPU. The model training takes about two to four weeks.

4. Experimental results

In this section, we validate the effectiveness of OCEAN approach through various experiments: First, we verify that the designed pretext task facilitates the learning of high-level semantics (Section 4.1). Secondly, we perform an ablation study to quantify the contributions of individual OCEAN components to the overall system performance (Section 4.2). Finally, we present the transfer learning on PASCAL VOC tasks and compare the proposed approach to other state-of-the-art methods (Section 4.3).

4.1. Model analysis

4.1.1. Fine-tuning AlexNet to the pretext task

In self-supervised learning, it is important to show that the designed pretext task requires high-level semantics. To verify that our pretext task, called pairwise classification, is related to semantic classification, we follow the experimental protocol of Noroozi and Favaro (2016). We copied the weights of the convolutional layers from the AlexNet that was trained for image classification using the ImageNet dataset (Krizhevsky et al., 2012). Specifically, to train our network, the weights for ith convolutional layers (conv < i>, (i = 1, 2, …, 5)) are copied from the AlexNet while the remaining layers are initialized to random Gaussian values where the mean and the standard deviation are set to 0 and 0.01, respectively. We trained our model with the use of the 100k images of the ILSVRC 2012 dataset, and measure the classification accuracy on PASCAL VOC 2007 test set by computing the hit ratio. As shown in Table 2, the results up to ‘Alex (>4)’ show similar classification performance. This demonstrates that the proposed pretext task is related to image classification that requires semantic
features. It is worth noting that there is a significant drop when the weights of layers conv1 to conv5 are transferred from the AlexNet. This is probably due to the fact that the weights of layer conv5 in the AlexNet may specialize in an image classification task (Nguyen, Yosinski, & Clune, 2015; Noroozi & Favaro, 2016). For the baseline result, we have performed the experiment on AlexNet using randomly initialized parameters. It scores 3.8% in the pairwise classification of our pretext text since semantics are not learned.

4.1.2. Filter responses

In order to gain a better understanding of the internal visual representations of the OCEAN approach, we obtained highly firing images of each filter in the conv5 layer. To this end, using PASCAL VOC 2012 as a test set, we compute the magnitude of the image neuron response for units in the conv5 layer. The images are then ranked based on firing scores and select the top five ones as shown in Fig. 5. We can see that the filters of our conv5 layers often capture semantically meaningful categories, without any manual annotation during the training phase. For example, channel 2 fires on airplane and channel 51 fires on car. This demonstrates that OCEAN learns semantic features by learning to arrange object proposals.

4.1.3. Network dissection

We applied network dissection to OCEAN to further interpret the deep visual representations and quantify their interpretability (Bau, Zhou, Khosla, Oliva, & Torralba, 2017). We assessed the interpretability of OCEAN using the Broden dataset\(^2\), which consists of various visual concepts such as scene, object, part, material, texture, and color (Bau et al., 2017). Note that all of these models use the AlexNet architecture and are tested at conv5. Fig. 6(a) demonstrates the result of network dissection across different models of self-supervised learning: Tracking (Wang & Gupta, 2015), Objectcentric (Gao et al., 2016), Audio (Owens, Wu, McDermott, Freeman, & Torralba, 2016), Moving (Agrawal, Carreira, & Malik, 2015), Colorization (Zhang et al., 2016), Puzzle (Noroozi & Favaro, 2016), Crosschannel (Zhang et al., 2017), Egomotion (Jayaraman & Grauman, 2015), Context (Doersch et al., 2015), and Frameorder (Misra et al., 2016). Note that the threshold of unique detectors is set to 0.04 in Fig. 6. Here, Tracking (Wang

---

& Gupta, 2015) shows the highest number of unique detectors.\(^3\) Interestingly, many part detectors emerge in OCEAN, which corresponds to our pretext task: OCEAN is trained to associate object parts considering their spatial relationships. It is thus demonstrated that the proposed pretext task commonly requires object parts to solve the problem.

Fig. 6(b) shows the interpretability of all five convolutional layers; ‘ours < i >’ symbolizes the ith convolutional layer of OCEAN. It is shown that OCEAN learns various semantics in all five convolutional layers and that it specifically captures object parts in an effective way. Fig. 7 presents visual detectors identified by OCEAN with high scores. It is observed that color and texture concepts are prevalent at lower layers ours1 and ours2 while more part detectors emerge in higher layers. Although ours2 in Fig. 6 shows high interpretability in parts, the highly activated part region shows texture-like regions as shown in the first image in Fig. 7(b). From Figs. 6 and 7, it is seen that OCEAN learns semantics through all convolutional layers with higher layers representing higher-level semantic concepts.

4.2. Analysis of the framework components

We analyzed three components of our framework: learning from non-overlapping patches, the effect of using object proposals instead of uniform patches, and proposals selection. To this end, the proposed pretext task, i.e., pairwise classification, was performed with alternative component configurations.

4.2.1. Non-overlapping patches

We tested the OCEAN framework without the ROI pooling layer, which is an architecture similar to that of Context (Doersch et al., 2015). Inputs are directly extracted from an image with respect to the size of object proposals and resized to the same size by means of interpolation. We used 50k and 250k images for training. Interestingly, for a training set of size 50k, the pairwise classification (Table 3) accuracy rate of Context and OCEAN is 30.1% and 65.5%, respectively. This can be an effect of the ROI pooling layer in the OCEAN framework, which generates features of the same size from conv5 layers regardless of the size of object proposals. Based on the results, we observe that OCEAN handles trivial shortcuts better than the baseline architecture (Doersch et al., 2015), since the pairwise classification requires high-level semantics in order to attain high accuracy as described in Section 4.1.

4.2.2. Object proposals vs. regular grid of patches

We tested OCEAN with nine regular patches that were randomly generated. To this end, we used 50k and 250k training images. For the pretext task, the results (pairwise classification in Table 4) of OCEAN with regular patches and object proposals were found to be 35.4% and 66.0%, respectively. This clearly demonstrates that object proposals are much more effective than naively using regular patches.

4.2.3. Handling object proposals

To verify the importance of selecting object proposals, we present illustrative results of a statistical analysis pertaining to object proposal selection. Proposal selection is required due to the potentially large set of candidates, amounts to hundreds and thousands per image (Uijlings et al., 2013). Table 5 compares the effect of handling object proposals. The experiments are performed using the ground-truth boxes in the ILSVRC 2012 dataset. In order to estimate how much of the ground-truth region is covered by the set "

---

\(^3\) As described in Bau et al. (2017), the interpretability does not always guarantee the discriminative power of the CNN models but a different quality that must be measured.

---

### Table 3

<table>
<thead>
<tr>
<th># Training data</th>
<th>Context</th>
<th>OCEAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>50k</td>
<td>30.1</td>
<td>65.5</td>
</tr>
<tr>
<td>250k</td>
<td>50.6</td>
<td>70.1</td>
</tr>
</tbody>
</table>

### Table 4

<table>
<thead>
<tr>
<th># Training data</th>
<th>Patches</th>
<th>Object proposals</th>
</tr>
</thead>
<tbody>
<tr>
<td>50k</td>
<td>35.4</td>
<td>66.0</td>
</tr>
<tr>
<td>250k</td>
<td>60.6</td>
<td>70.8</td>
</tr>
</tbody>
</table>

---
of input proposals, the average precision (AP) between the ground-truth box and the varying numbers of input proposals is computed. This is important to OCEAN because higher AP indicates that potential objects can be exploited as inputs more frequently during the self-supervised learning, enabling representations of high-level semantics.

Table 5 shows that input proposals commonly score higher APs when object proposals are filtered, which demonstrates that input proposals (with filtering) commonly cover a larger object region. Illustrative results in Fig. 8 also demonstrate the effectiveness of object proposal selection. Due to the characteristics of the object proposals, inputs can be extracted from similar regions (puppies in Fig. 8(a)) which can be a distraction for the CNN in their attempt to capture the potential objects effectively. It is also prone to generate less meaningful inputs (a top left figure in Fig. 8(a)) which do not cover the object region effectively.

We further compare our classification task with and without using a proposal selection strategy as described in Section 3.3. In both cases, we train CNN with 50k, 100k, and 250k images sampled from the ILSVRC 2012 train dataset. For testing the classification performance, we use the PASCAL VOC 2007 test set.
Table 6 shows that enhanced performance can be achieved by filtering object proposals according to the proposed approach. This confirms that the selection of appropriate proposals is critical to learning various types of geometric relationships.

We have presented the analysis of the proposed model by changing configurations. The results from Sections 4.2.1. and 4.2.2. demonstrate that the proposed model outperforms previous approaches which employ regular size and non-overlapping patches (Doersch et al., 2015; Noroozi & Favaro, 2016). This shows that the proposed model with the object proposals and ROI pooling generates less clutter and provides effective learning rather than simply using non-overlapping regular patches. Although the proposed approach requires an additional off-line process to generate object proposals, it helps better understanding of images for learning the high degree of visual representations.

4.3. Transfer learning on PASCAL VOC tasks

We compared ours with other self-supervised learning methods (Agrawal et al., 2015; Doersch et al., 2015; Lee et al., 2017; Noroozi & Favaro, 2016; Pathak et al., 2017; Pathak et al., 2016; Wang & Gupta, 2015; Zhang et al., 2016; 2017) by fine-tuning the pre-trained models to the PASCAL VOC 2007 and VOC 2012 datasets. We evaluated OCEAN in the object detection (Girshick, 2015), classification (Krähenbühl, Doersch, Donahue, & Darrell, 2015), and action recognition tasks (Gkioxari et al., 2015). AlexNet (Krizhevsky et al., 2012) is a base model for all methods, while there exist slight differences regarding the use of batch normalization (Doersch et al., 2015) and the size of the stride (Noroozi & Favaro, 2016). We copy the weights of conv1 to conv5 layers of our pre-trained model and initialize subsequent layers for each task with Gaussian random weights that have a mean of 0.1 and a standard deviation equal to 0.001.

4.3.1. Object detection

We compared our approach with the framework of Fast-RCNN (Girshick, 2015) for object detection using the PASCAL VOC 2007 and 2012 datasets. In the testing, the average of the object detection time on the PASCAL VOC datasets is recorded to 0.22s per image. Following the published guidelines in Pathak et al. (2016), Doersch et al. (2015), Noroozi and Favaro (2016) and Wang and Gupta (2015), we perform the fine-tuning on trainval and train sets in VOC 2007 and VOC 2012, respectively, and test the performance on the test and val sets, respectively. We use the multi-scale strategy for training and testing as described in Girshick (2015).

4.3.2. Image classification

We experimented with image classification on PASCAL VOC 2007 (object categories). We follow the baseline of Krähenbühl et al. (2015), using random crops during training and average scores from 10 crops during testing. Training and testing are performed on trainval and test sets, respectively. The average of processing time in action recognition is recorded to 0.08s per image.

4.3.3. Action recognition

We experimented with the action-specific framework, R-CNN (Gkioxari et al., 2015). All the pre-trained models were fine-tuned on PASCAL VOC Action 2012 dataset (Everingham, Van Gool, Williams, Winn, & Zisserman, 2010). Training and testing are performed on train and val sets, respectively. The average of processing time in action recognition is recorded to 0.46s per image.

4.3.4. Performance analysis

Table 7 summarizes the fine-tuning results in object detection and classification. Results from other methods were taken from Pathak et al. (2016), Noroozi and Favaro (2016), Pathak et al. (2017) and Lee et al. (2017). All models have a CNN architecture similar to AlexNet, but differ in minor details such as the presence of batch normalization layers, stride, or the presence of grouped convolutions. Note that Noroozi et al. (Noroozi & Favaro, 2016) use a different size of the network, changing stride 4–2 in conv1 layers, and produce better results but this is not a fair comparison to other methods.

The Tracking (Wang & Gupta, 2015) shows comparable performance to our approach in the classification task, but not in object detection. Impainting (Pathak et al., 2016) achieves the

### Table 6
Comparison of the pairwise classification scores (in percentage) with and without filtering object proposals.

<table>
<thead>
<tr>
<th># Training data</th>
<th>No-filtering</th>
<th>Filtering</th>
</tr>
</thead>
<tbody>
<tr>
<td>50k</td>
<td>63.5</td>
<td>65.5</td>
</tr>
<tr>
<td>100k</td>
<td>65.0</td>
<td>68.5</td>
</tr>
<tr>
<td>250k</td>
<td>70.6</td>
<td>72.1</td>
</tr>
</tbody>
</table>

### Table 7
Fine-tuning pre-trained models on PASCAL VOC dataset. Average precisions (APs) are evaluated for each task.

<table>
<thead>
<tr>
<th>Method</th>
<th>Pre-training</th>
<th>Supervision</th>
<th>PASCAL VOC 2007</th>
<th>PASCAL VOC 2012</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Classification</td>
<td>Detection</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supervised learning</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AlexNet</td>
<td>3 days</td>
<td>1000 class labels</td>
<td>78.2</td>
<td>56.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unsupservised learning</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tracking (Wang &amp; Gupta, 2015)</td>
<td>1 week</td>
<td>Motion</td>
<td>58.4</td>
<td>44.0</td>
</tr>
<tr>
<td>Moving (Agrawal et al., 2015)</td>
<td>–</td>
<td>Motion</td>
<td>52.9</td>
<td>41.8</td>
</tr>
<tr>
<td>Motionseg (Pathak et al., 2017)</td>
<td>–</td>
<td>Motion</td>
<td>56.5</td>
<td>44.5</td>
</tr>
<tr>
<td>GPN (Lee et al., 2017)</td>
<td>3 days</td>
<td>Motion</td>
<td>63.8</td>
<td>48.9</td>
</tr>
<tr>
<td>Colorization (Zhang et al., 2016)</td>
<td>–</td>
<td>Color</td>
<td>65.9</td>
<td>46.9</td>
</tr>
<tr>
<td>Crosschannel (Zhang et al., 2017)</td>
<td>–</td>
<td>Reconstruction</td>
<td>67.1</td>
<td>46.7</td>
</tr>
<tr>
<td>Inpainting (Pathak et al., 2016)</td>
<td>14 h</td>
<td>Context</td>
<td>56.5</td>
<td>44.5</td>
</tr>
<tr>
<td>Context (Doersch et al., 2015)</td>
<td>4 weeks</td>
<td>Context</td>
<td>55.3</td>
<td>46.6</td>
</tr>
<tr>
<td>Puzzle (Noroozi &amp; Favaro, 2016)</td>
<td>2.5 days</td>
<td>Context</td>
<td>68.6</td>
<td>51.8</td>
</tr>
<tr>
<td>OCEAN-2</td>
<td>2 weeks</td>
<td>Context</td>
<td>58.8</td>
<td>47.5</td>
</tr>
<tr>
<td>OCEAN-4</td>
<td>4 weeks</td>
<td>Context</td>
<td>60.0</td>
<td>48.6</td>
</tr>
</tbody>
</table>

Note that Puzzle (Noroozi & Favaro, 2016) is performed under slightly different convolutional architecture with a finer stride at conv1, preventing fair comparisons.
fastest training time, but does not work well for all tasks. Context (Doersch et al., 2015) exhibits competitive performance, but it requires considerable training time. Context learns to discover relative positions between non-overlapping patches. These non-overlapping patches may include clutter, which is not helpful for training. Overall, existing methods yield competitive performance in a specific task. In contrast, OCEAN-2 and OCEAN-4 generally produce comparable results to other methods in object detection as well as in image classification. Note that OCEAN-2 and OCEAN-4 denote the proposed approaches trained for 2 weeks and 4 weeks, respectively.

Since OCEAN might be biased to specific object proposals, we have performed the additional experiments on the PASCAL VOC 2007 detection task using three different proposals: Randomized-Prim (RP) (Manen et al., 2013), Multiscale Combinatorial Grouping (MCG) (Arbeláez, Pont-Tuset, Barron, Marques, & Malik, 2014), and Geodesic Object Proposals (GOP) (Krähenbühl & Koltun, 2014). As shown in Table 8, RP, MCG, GOP, and SelectiveSearch (SS) achieve a mAP of 48.8, 49.5, 48.8, and 48.6, respectively. This shows that the performance of fine-tuning results of the OCEAN is not biased with respect to the object proposal methods.

4.3.5. Changing the number of fine-tuning layers

We have experimented on PASCAL VOC 2012 detection and action recognition by changing fine-tuned layers as in Pathak et al. (2017). In object detection shown in Table 9, configurations labeled as ‘>-0’, ‘>-1’, and ‘>-2’, exhibit favorable performance while others decrease with large degrees. Since OCEAN captures object parts in high level convolutional layers, the detection performance can decrease if the high level convolutional layers are frozen. Contrary to the above, in action recognition tasks reported in Table 10, high performance is preserved in every case.

From the experiment on the PASCAL VOC dataset, we can verify that OCEAN contains a high degree of semantics which helps learning vision tasks such as object detection, classification, and action recognition. Furthermore, noting that OCEAN outperforms in the action recognition, we can analyze the results by taking the observations from Fig. 6(b). The semantics in OCEAN adapt well to the R-CNN (Gkioxari et al., 2015) which considers contextual information during training the network. This resembles the idea of OCEAN which naturally considers the relationship between objects and parts during training the network.

5. Feature representations

In this section, we evaluate the generalization capability of the OCEAN. We consider OCEAN as a generic feature descriptor and apply it to other applications: image retrieval and semantic correspondences (Ham et al., 2016).

5.1. Image retrieval

We analyze the learned visual representations by using an image retrieval task of the PASCAL VOC 2007 dataset. Query and test images are extracted from the PASCAL VOC 2007 training and test datasets, respectively, using the provided ground-truth bounding boxes for each label. We extract conv5 features of the query and target images and measure their similarity using the dot product of the conv5 outputs.

In order to obtain quantitative results, we measure the retrieval rate by counting the number of correct retrievals of the top 20 retrieved images. The retrieval is regarded as being correct if the classes of the target and the query images are the same. We considered three models: AlexNet trained with ImageNet classification (Krizhevsky et al., 2012), AlexNet initialized with random weights, and our proposed model. Noting that all features from images are extracted offline, the runtime for each model shows 3.7s on average. In experiments, we obtained a retrieval rate of 67.2% for AlexNet which is a supervised pre-trained model. Our pre-trained model yielded a retrieval rate equal to 43.5%, which is higher than the 28.9% retrieval rate of the AlexNet with random weights. Fig. 9 shows illustrative results of top 5 retrievals for query images. It can be seen that OCEAN estimates objects with similar shapes satisfactorily and often captures the correct category while being trained without any annotated class labels. OCEAN captures semantic features from low-resolution images as shown in the examples of the monitor and the airplane in Fig. 9(d). This demonstrates that OCEAN has the capability of capturing semantic features from inputs of various sizes.

5.2. Semantic correspondences

We apply OCEAN to establish semantic correspondences. Semantic correspondence methods are designed to handle images depicting different instances of the same object or scene category. We use a proposal flow (PF) method (Ham et al., 2016) that estimates correspondences by matching object proposals. Experiments are performed based on the PF benchmark (Ham et al., 2016), which provides the evaluation metrics and 10 object classes. We use two feature descriptors: outputs of conv4 and conv5 of AlexNet (Krizhevsky et al., 2012) (Alex_c4 and Alex_c5) and ours (OCEAN_c4 and OCEAN_c5). The output of conv5 with random weights (Rand) is also considered. The runtime for each AlexNet-like architecture shows 11.7s on average.

The matching precision and retrieval performance are measured by computing probability of correct region (PCR) and an average of IoU of k-best matches (miIoU@k).

Fig. 10 presents illustrative results of the PF dataset. Despite the fact that it is trained in an unsupervised manner, the OCEAN aligns semantically similar, but not identical, images well. As shown in

Table 8

<table>
<thead>
<tr>
<th>Methods</th>
<th>mAP(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP (Manen et al., 2013)</td>
<td>48.8</td>
</tr>
<tr>
<td>MCG (Arbeláez et al., 2014)</td>
<td>49.5</td>
</tr>
<tr>
<td>GOP (Krähenbühl &amp; Koltun, 2014)</td>
<td>48.8</td>
</tr>
<tr>
<td>SS (Oijlings et al., 2013)</td>
<td>48.6</td>
</tr>
</tbody>
</table>

Table 9

PASCAL VOC 2012 object detection. (‘>-’: layers above convi are fine-tuned; ‘>-0’: the entire net is fine-tuned.)

<table>
<thead>
<tr>
<th>Methods</th>
<th>mAP(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inpainting</td>
<td>39.1</td>
</tr>
<tr>
<td>Context</td>
<td>47.5</td>
</tr>
<tr>
<td>Puzzle</td>
<td>49.0</td>
</tr>
<tr>
<td>Motionseg</td>
<td>48.6</td>
</tr>
<tr>
<td>OCEAN-4</td>
<td>49.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>mAP(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inpainting</td>
<td>30.9</td>
</tr>
<tr>
<td>Context</td>
<td>48.1</td>
</tr>
<tr>
<td>Puzzle</td>
<td>24.3</td>
</tr>
<tr>
<td>OCEAN-4</td>
<td>54.2</td>
</tr>
</tbody>
</table>

Table 10

PASCAL VOC 2012 action classification. (‘>-’: layers above convi are fine-tuned; ‘>-0’: the entire net is fine-tuned.)

<table>
<thead>
<tr>
<th>Methods</th>
<th>mAP(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inpainting</td>
<td>39.1</td>
</tr>
<tr>
<td>Context</td>
<td>47.5</td>
</tr>
<tr>
<td>Puzzle</td>
<td>49.0</td>
</tr>
<tr>
<td>OCEAN-4</td>
<td>49.4</td>
</tr>
</tbody>
</table>
Fig. 9. Image retrieval results. (top) Query images and (bottom) the top-5 retrieval results of (left) AlexNet trained with ImageNet labels, (middle) AlexNet initialized with random variables, and (right) ours. The OCEAN captures high-level semantics while it is employed as the CNN-based feature descriptor.

Fig. 10. Results of the semantic correspondence matching. (a–b) Source images are warped to the target images using dense correspondences estimated by Ham et al. (2016), using conv5 features of (c) AlexNet trained with ImageNet labels, (d) AlexNet initialized with random variables, and (e) ours.

Fig. 11, OCEAN exhibits better performance compared to the randomized CNN, a result which demonstrates that semantic features are learnt from the pretext task. Although OCEAN has inferior performance compared to AlexNet, a similar performance trend is observed in both models; the conv4 features exhibit slightly better performance compared to the conv5 features. Comprehensive performance evaluation and comparison to state-of-the-art self-supervised learning methods demonstrates that OCEAN can learn high-level visual semantic representations without human annotation.

6. Conclusion and future work

We have presented OCEAN, a CNN framework for a self-supervised learning of visual representations. We have employed the idea of a pretext task which is used for the purposes of teaching the CNN to discover the geometric relationships between a pair of object proposals. The OCEAN enables CNN to identify potential objects and their parts during self-supervised learning. In a series of experiments, we have performed various ablation studies to verify that OCEAN learns high-level semantics. The results have shown that OCEAN alleviates trivial shortcuts and learns high-level semantics from object proposals. Furthermore, application of OCEAN to fine-tuning on PASCAL VOC object detection, classification, and action recognition tasks, demonstrates the competitive performance compared to other unsupervised learning methods. Performance evaluation has also shown that OCEAN can be used as a generic feature extractor by applying it to image retrieval and semantic matching.

While many approaches have commonly focused on training supervised manner, we have shown that OCEAN can effectively embed high degrees of semantic visual representations without human annotations. This is important in terms of intelligent systems since many problems require large-scale datasets to learn the features, such as action recognition for wearable sensors (Nweke et al., 2018) and data classification in the web (Aghamaleki & Baharlou, 2018). The self-supervised approach can handle the impediment caused by the supervised learning.

Future directions for this work can be described as follows:

- Multi-task self-supervised learning: We may improve the performance of our methods further by designing a novel loss function which helps to improve the CNN in learning rich high-level semantics and constructing a joint learning task. For example, OCEAN does not currently consider the relative feature distance between object proposals; it reveals geometric relationships between object proposals by means of softmax classification. We
believe that embedding a relative feature distance by a triplet loss function (Gao et al., 2016; Lee et al., 2017; Wang & Gupta, 2015) within a single image can achieve this with a subsequent improvement in performance.

- Task-oriented self-supervised learning: We can design the self-supervised learning for a specific computer vision application. E.g., the proposed model shows good results in action classification task since the feature from OCEAN is learned by discovering the relationship between object proposals, which is similar to the learning strategy in R* CNN (Gkioxari et al., 2015). The study on self-supervised approach for the specific application thus can be a good direction for further research.

- Learning visual representations from different dataset: Commonly the ImageNet dataset includes well-taken photos. However, in real world scenario, many photos may not be effective for self-supervised learning, e.g., only small objects with background regions. This can be another challenging issue to learn meaningful semantics from images.

- Application to intelligent systems: The self-supervised approach also can be combined with visual navigation for autonomous vehicles (Charalampous, Kostavelis, & Gasteratos, 2016). The network can learn meaningful features during visual navigation in a self-supervised manner, enabling to generate an effective decision for navigating the environment.

Credit authorship contribution statement

Changjae Oh: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Visualization, Writing - original draft, Resources, Writing - review & editing. Bumsub Ham: Conceptualization, Investigation, Methodology, Writing - original draft, Writing - review & editing, Validation. Hansung Kim: Investigation, Methodology, Writing - review & editing, Validation. Adrian Hilton: Supervision, Writing - review & editing, Validation. Kwanghoon Sohn: Funding acquisition, Project administration, Supervision, Writing - review & editing.

References