Self-Supervised Representation Learning via Neighborhood-Relational Encoding

Mohammad Sabokrou
Institute for Research in Fundamental Sciences
sabokro@ipm.ac.ir

Mohammad Khalooei
Amirkabir University of Tech.
khalooei@aut.ac.ir

Ehsan Adeli
Stanford University
eadeli@cs.stanford.edu

Abstract

In this paper, we propose a novel self-supervised representation learning by taking advantage of a neighborhood-relational encoding (NRE) among the training data. Conventional unsupervised learning methods only focused on training deep networks to understand the primitive characteristics of the visual data, mainly to be able to reconstruct the data from a latent space. They often neglected the relation among the samples, which can serve as an important metric for self-supervision. Different from the previous work, NRE aims at preserving the local neighborhood structure on the data manifold. Therefore, it is less sensitive to outliers. We integrate our NRE component with an encoder-decoder structure for learning to represent samples considering their local neighborhood information. Such discriminative and unsupervised representation learning scheme is adaptable to different computer vision tasks due to its independence from intense annotation requirements. We evaluate our proposed method for different tasks, including classification, detection, and segmentation based on the learned latent representations. In addition, we adopt the auto-encoding capability of our proposed method for applications like defense against adversarial example attacks and video anomaly detection. Results confirm the performance of our method is better or at least comparable with the state-of-the-art for each specific application, but with a generic and self-supervised approach.

1. Introduction

The widespread adoption of deep learning methods in computer vision owes its success to learning powerful visual representations [3]; however, this was achievable only with intensive manual labeling effort (which is extravagant and not scalable). Therefore, unsupervised feature learning [7, 16, 29, 30, 37, 38, 59, 61, 64] has recently been widely adopted to extract data representation without the need for such label information. This representation can be used for different tasks of image [26] or video classification [23].

Unsupervised representation learning in the context of deep networks has often been defined by minimizing the reconstruction error [34], such as in auto-encoders (AEs). AEs have shown to be great tools for unsupervised representation learning in a variety of tasks, including image inpainting [43], feature ranking [54], denosing [57], clustering [65], defense against adversarial examples [35], and anomaly detection [48, 52]. Although AEs have led to far-reaching success for data representation, there are some caveats associated with using reconstruction errors as the sole metric for representation learning: (1) As also argued in [58], it forces to reconstruct all parts of the input, even if they are irrelevant for any given task or are contaminated by noise; (2) It leads to a mechanism that entirely depends on single-point data abstraction, *i.e.*, the AE learns to just reconstruct its input while neglecting other data points present

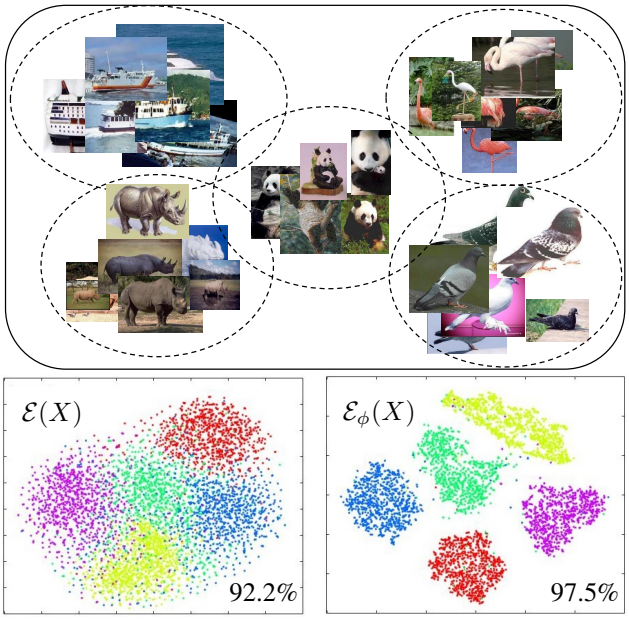

Figure 1. Some samples from five classes of the Caltech dataset (top), latent space visualization using a regular AE (*i.e.*, $\mathcal{E}(X)$; left), and our proposed AE that encodes the neighborhood relations (*i.e.*, $\mathcal{E}_\phi(X)$; right). With the same classifier, $\mathcal{E}(X)$ leads to a classification accuracy of 92.2% and $\mathcal{E}_\phi(X)$ 97.5%.

in the dataset. The semantic relationship between neighboring samples in the dataset endure rich information that can direct learning more representative features (overlooked in current AE settings).

To overcome the above challenges and enhance the performance of the popular encoder-decoder networks (*i.e*., AEs), in this paper, we propose a simple yet effective encoder-decoder architecture using self-supervised learning strategies. The self-supervised component encodes the neighborhood relations among the data points present in the training set. This setting goes beyond looking at the reconstruction of each data point separately, and self-supervises the model such that the conceived latent space preserves proper local neighborhood structures. Different from most previous works [20, 34], which aim at preserving the global Euclidean structure, our proposed Neighborhood-Relational Encoding (NRE) aims at preserving the local neighborhood structure on the data manifold. As a result, we expect that NRE will be less sensitive to noise and outliers. Our proposed structure includes an encoder that also encodes neighborhood relations, denoted by $\mathcal{E}_\phi$ (as opposed to $\mathcal{E}$ in regular AEs), and a decoder, $\mathcal{D}$, which are jointly learned (similar to an auto-encoder). Therefore, $\mathcal{E}$ encodes the input sample X to a discriminative latent space $\mathcal{R}$, from which $\mathcal{D}$ must be able to retrieve the original sample. To learn the neighborhood relations, $\mathcal{E}_\phi$ requires to operate as a kernel [18] and map close-by data points closely to each other in the latent space (see Fig. 1).

In summary, the main contributions of this paper are as follows: (1) We propose a new learning strategy for $\mathcal{E}$ncoder-$\mathcal{D}$ecoder deep network by introducing NRE. To the best of our knowledge, this article is the first to present an encoding network that simultaneously learns kernels (neighborhood cues) among inputs. (2) Leveraging the self-supervision injected as a result of our NRE component, we improve the performance of auto-encoders, which are popular tools of feature learning, (3) Our proposed scheme efficiently learns the semantic concepts within the visual data, and achieves state-of-the-art results on different applications, such as image classification, anomaly detection, and defense against attacks from adversarial examples.

2. Related Work

Unsupervised learning through learning representation space that successfully reconstructs samples is widely used for a variety of tasks, including classification [26], de-nosing [57], and in-painting [43]. Conventional methods for unsupervised representation learning are usually based on a pretext task such as reconstruction of static images [38] or videos [59]. Learning to reconstruct data was used for tasks like de-nsoing [57], in-painting [42], image refinement for defense against adversarial example [53], and for one-class classifiers [46, 51, 52]. This paper focuses on a new way for training encoder-decoder networks incorporating self-supervised neighborhood constraints. In the following, we briefly survey recent un/self-supervised representation learning and learn-to-reconstruct methods.

Un/Self-Supervised Representation Learning: Learning with respect to a pretext task is the central idea for unsupervised representation learning. As mentioned, learning-to-reconstruct images is a common pretext task for unsupervised feature learning [20]. Earlier works were based on precisely reconstructing the input images. But recent work tried constructing other modes of the data alongside reconstructing images themselves. Some examples include constructing an image channel from another one [64], colorizing gray-scale images [27, 63], and in-painting [43].

Other types of pretext tasks proposed for unsupervised learning include understanding the correct order of video frames [6, 36] or predicting the spatial relation between image patches [12], *e.g*., jigsaw puzzle solving as a pretext task was exploited by Noorozi and Favaro [37]. In another work, Noroozi *et al*. [38] proposed to train an unsupervised model by counting the primitive elements of images. Pathak *et al*. [41] proposed a model to segment an image into foreground and background. Some methods use external signals that may come freely with visual data. For instance, some methods use known motion cues like ego-motion [2, 21] or sound [39] as sources for self-supervision. Most of these works ignored the relationship between samples. Some recent work [2, 41] tried to model the relation between video patches as a pretext task. Conceptually, these works are related to our work, but different from our method, these pre-trained networks were developed for ad-hoc purposes, and were not capable of being applied to other computer vision tasks. Additionally, we introduce more comprehensive neighborhood cues to discover the intrinsic local manifolds. Che *et al*. [9] proposed a method for unsupervised feature learning using a similarity-aware auto-enocder that aims to map similar samples close to each other. However, unlike our method, they neglected the important relational information among the samples.

Learning-to-Reconstruct: As discussed earlier, reconstruction can be considered as a pretext task for un/self-supervised representation learning. Many of computer vision tasks are dependent on this simple idea. There is a wide range of applications, but we briefly go over the tasks used for evaluation in this paper.

Sabokrou *et al*. [51, 52] used reconstruction errors and the reconstructed video frames for end-to-end one-class classification applied to anomaly detection. They analyzed the reconstruction error for detecting anomalies [46] the reconstructed (or refined) images to create better discrimination between normal and anomaly images [51]. MagNet [35] and Defense-GAN [53] as two important baseline for defense against adversarial attacks are based on refinement

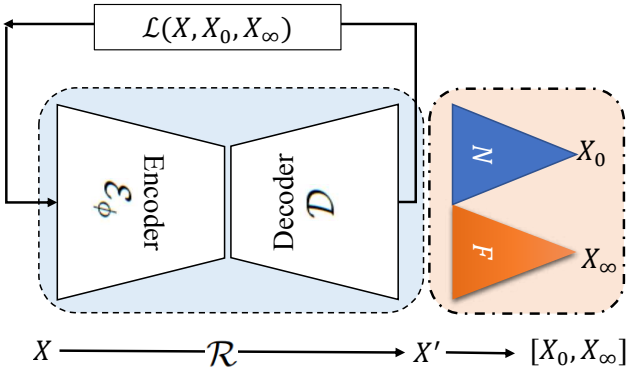

Figure 2. Overview of the proposed structure for self-supervised feature learning. $\mathcal{E}+\mathcal{D}$ are learned through a forward path and back-propagation of the error. In the forward path, X' is retrieved from the input X, but a relational loss ($\mathcal{L}(X, X_0, X_\infty)$) is back-propagated to train the $\mathcal{E}+\mathcal{D}$ networks. N and F are the modules that identify X_0 and X_∞ from the dataset.

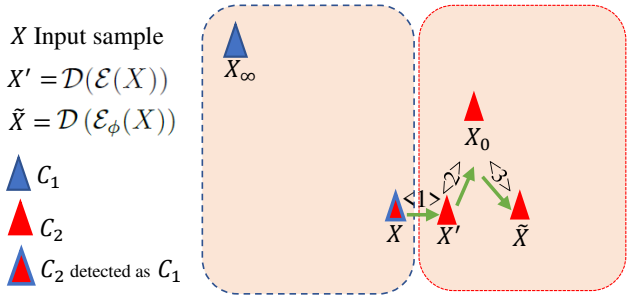

Figure 3. A schematic sketch of procedures for reconstructing X in 2D space. Suppose we have two classes of data (blue C_1 and red C_2). Let $X \in C_2$, but being mistakenly classified as C_1. Here, we analyze and apply terms of the loss function in (1) one by one. Based on term Eq. (1), X is supposedly transferred to (reconstructed as) X' using <1>; the second and third terms (<2> and <3>) bring X' closer to X_0 and farther from X_∞, respectively. As can be seen, the final reconstructed sample ($\tilde{X} = \mathcal{D}(\mathcal{E}_\phi(X))$) may be put on the correct side.

of adversarial examples using reconstruction techniques. MagNet directly refines the adversarial examples using an encoder-decoder trained on normal samples. Defense-GAN refines the adversarial example using a GAN generator that is trained only on normal images. The generator maps input examples to its latent space and generates images from the latent space that are hopefully free from the adversary.

3. Method

Our proposed approach for self-supervised representation learning is composed of three important components: (1) The encoder network $\mathcal{E}$; (2) the decoder network $\mathcal{D}$; and (3) the objective function that incorporates the neighborhood relational information. The joint network $\mathcal{E}+\mathcal{D}$ is trained as an encoder-decoder network based on the proposed objective function. $\mathcal{E}$ provides a reduced representation $\mathcal{R}$ of its input sample X, with maximum information preservation, which enables $\mathcal{D}$ to retrieve X from $\mathcal{R}$. The output of $\mathcal{D}$ is denoted by $\hat{X}$. Our goal is to train this reconstruction so that $\hat{X}$ is similar not only to X but also to its neighbouring sample(s) X_0, while being dissimilar to its far-away sample(s), *i.e.*, X_∞. This infrastructure concludes a self-supervised (and hence unsupervised) representation ($\mathcal{E}(X)$) that can be used for any image or video analysis tasks. Encoding the neighborhood relations into the representation makes the learned feature space more separable. Fig. 2 shows a sketch of our approach. $\mathcal{E}$ and $\mathcal{D}$ are trained to discover the relationship among the samples.

First, consider a setting that $\mathcal{E}+\mathcal{D}$ defines an auto-encoder (AE) that is pre-trained to only reconstruct the input sample, *i.e.*, reconstruct X and obtain X'. Using this pre-trained network, we propose a procedure to identify X_0 and X_∞ based on the latent space of $\mathcal{E}+\mathcal{D}$, $\mathcal{R}$, using the two modules, N and F, respectively. Then, we optimize the network parameters of $\mathcal{E}$ and $\mathcal{D}$ jointly using a loss function $\mathcal{L}(X, X_0, X_\infty)$, in which the neighborhood-relational information is propagated with respect to sample X. We denote our AE encoder that incorporates neighborhood information for building the latent space as $\mathcal{E}_\phi$. As the training continues, $\mathcal{E}_\phi+\mathcal{D}$ learns to better encode neighborhood information into $\mathcal{R}$ and hence N and F can better uncover close-by and far-away samples. After training the network, $\mathcal{R}$ (*i.e.*, $\mathcal{E}_\phi(X)$) provides a discriminative representation of X. Furthermore, $\mathcal{D}(\mathcal{E}_\phi(X))$ acts as a refiner for X regularizing the reconstructed X' by its neighbors, which can be integrated as a pre-processing step for different classification or regression tasks. X' is the reconstructed version of X in the training phase, while $\tilde{X}$ is its reconstructed version using the trained relational AE. Detailed descriptions of each module and the overall training/testing procedures are described in the following subsections.

3.1. Neighborhood-Relational Encoding (NRE)

Traditional unsupervised representation learning highly depends on a pretext assumption, often defined on top of the reconstruction power of the learned features. These methods learn the spatial dependencies within the images and hence the inter-relations among the data points are neglected. As shown by methods that operate in the neighborhood spaces, such as K-nearest neighbour [10], as a rule of thumb, samples that are spanned close-by in the space of all samples tend to belong to the same classes. Also, kernel methods [55] suggest that modifying the representation space based on the (positive-definite) similarities generally leads to more dicriminative and separable spaces. Encoder-decoder networks have also been investigated for such properties, *e.g.*, in [51] where it was shown that samples can be

efficiently refined and be made more separable for anomaly classification tasks. Inspired by the previous work, we propose an encoder-decoder AE deep network to learn representations of the data through self-supervision derived from the neighborhood cues in the data manifold.

To this end, we force $\mathcal{D}\left(\mathcal{E}_\phi(X)\right)$ to reconstruct X mindful of its neighbor(s) X_0, while trying to distant from the far-away sample(s) X_∞. Therefore, the parameters of $\mathcal{E}_\phi + \mathcal{D}$ are learned using the following loss function:

$$\begin{aligned}\mathcal{L} = &\underbrace{\lambda_1 \mathbb{D}(\mathcal{R}_\mathcal{A}(X), \mathcal{R}_\mathcal{A}(X'))}_{<1>} + \underbrace{\lambda_2 \mathbb{D}(\mathcal{R}_\mathcal{A}(X'), \mathcal{R}_\mathcal{A}(X_0))}_{<2>} \\ &+ \underbrace{\lambda_3 \mathbb{S}(\mathcal{R}_\mathcal{A}(X'), \mathcal{R}_\mathcal{A}(X_\infty))}_{<3>},\end{aligned} \tag{1}$$

where $X' = \mathcal{D}(\mathcal{E}(X))$, $\lambda_{i\in\{1,2,3\}}$ are scaled regularization hyperparameters with $\lambda_1 + \lambda_2 + \lambda_3 = 1$, X_0 and X_∞ are calculated by $N(X)$ and $F(X)$, respectively. $N(\cdot)$ and $F(\cdot)$ define functions that return the closest and the farthest samples to their inputs, respectively (defined in detail later). $\mathbb{D}$ and $\mathbb{S}$ are two metrics for computing the distance and similarity of two vectors, and $\mathcal{E}_\phi(X)$ is the new representation of X. To obtain the similarities and distances of samples we use a pre-trained AE network. We denote the encoder of this AE as $\mathcal{A}$ and its latent space as $\mathcal{R}_\mathcal{A}$. $\mathbb{S}$, $\mathbb{D}$, and $\mathcal{R}_\mathcal{A}$ are explained in more details in the following subsections. After training this network, $\mathcal{R} = \mathcal{E}_\phi(X)$ will be a representation of X, which is forced be similar to the representation of the closest sample(s) and dissimilar to far-away one(s). Fig. 3 shows how sample X is refined in a 2D space with respect to each term in the loss function, Eq. (1). As can be seen, after refining, X moves closer to the center of its correct class. Note that X_0 and X_∞ should be calculated with respect to X', not X.

Neighboring Relation As shown in Fig. 2, there are two important modules, D and F, with key roles that provide side information for joint training of $\mathcal{E}_\phi + \mathcal{D}$. As mentioned earlier, D and F are defined to find the closet or most similar sample(s) and far-away or most dissimilar sample(s), respectively. There are several measures to infer the similarity of two samples (*e.g.*, images) in an unsupervised fashion. Direct image similarity methods, such as SSIM [60], are too high-level and often fail to evaluate the semantics of the image. Hence, instead of directly working with images in the original space, we compare them in the latent representation space. To this end, an encoder network, $\mathcal{A}$, is trained on all unlabeled available samples to provide a discriminative representing of the samples. The encoder unsupervisedly and jointly with a decoder, is trained to form an auto-encoder. Let $\mathcal{X} = \{X_i\}_{i=1}^{i=Z}$ be our dataset with size Z and $\mathcal{R}_\mathcal{A}(X_i)$ be the corresponding representation on X_i using $\mathcal{A}$.

Under the above setting, X_0, the closest sample to X' is calculated using

$$X_0 = N(X') = \underset{X_i \in \mathcal{X}, X_i \neq X'}{\operatorname{argmax}} \mathbb{S}(\mathcal{R}_\mathcal{A}(X_i), \mathcal{R}_\mathcal{A}(X')), \tag{2}$$

and X_∞, the most dissimilar sample to X is defined as:

$$X_\infty = F(X') = \underset{X_i \in \mathcal{X}, X_i \neq X'}{\operatorname{argmin}} \mathbb{S}(\mathcal{R}_\mathcal{A}(X_i), \mathcal{R}_\mathcal{A}(X')). \tag{3}$$

In addition, $\mathbb{D}(\cdot,\cdot) = 1 - \mathbb{S}(\cdot,\cdot)$ and $\mathbb{S}(\cdot,\cdot)$ is a cosine similarity measure computed by:

$$\mathbb{S}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}\cdot\mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|} = \frac{\sum_{i=1}^n a_i b_i}{\sqrt{\sum_{i=1}^n a_i^2}\sqrt{\sum_{i=1}^n b_i^2}}, \tag{4}$$

where a_i and b_i are the i^{th} components of vectors $\mathbf{a}$ and $\mathbf{b}$, respectively. Without loss of generality, we can select a set of (more than one) similar and dissimilar samples. $\mathcal{A}$ is composed of several convolutional, sub-sampling, and fully connected layers There is a ReLU layer on top of $\mathcal{R}_\mathcal{A}$, which forces all of its output values to be positive. Consequently, always $\mathbb{S}(\mathcal{R}_\mathcal{A}(X_i), \mathcal{R}_\mathcal{A}(X'))$ is positive.

3.2. Training $\mathcal{E}_\phi + \mathcal{D}$

These two networks are jointly trained. Training sample X is fed to $\mathcal{E}_\phi$+$\mathcal{D}$, which creates an output X'. The network is optimized using the loss function $\mathcal{L}$ (Eq. (1)). This turns X to a more discriminative sample based on the neighborhood encoding scheme (see Subsection 3.1). Eq. (1) considers only one nearby and one far-away sample, but for robustness against outliers and to better discover the relationship of samples, the network can be trained using a set of such samples (more than one). Therefore, the loss function could be re-written as follow:

$$\begin{aligned}\mathcal{L} = &\lambda_1 \mathbb{D}(\mathcal{R}_\mathcal{A}(X), \mathcal{R}_\mathcal{A}(X')) \\ &+ \lambda_2 \sum_{i=1}^{T} \mathbb{D}(\mathcal{R}_\mathcal{A}(X'), \mathcal{R}_\mathcal{A}(X_{0i})) \\ &+ \lambda_3 \sum_{i=1}^{T} \mathbb{S}(\mathcal{R}_\mathcal{A}(X'), \mathcal{R}_\mathcal{A}(X_{\infty i})),\end{aligned} \tag{5}$$

where T is a hyperparameter denoting the number of selected nearby/faraway samples. Generally, greater T concludes better performance, but its side-effect is an expensive training phase, and if a very large T is selected, then the set of far-away and nearby samples may have common elements, which is not desirable. Note that finding X_0 and X_∞ is a time consuming task, which is proportional to the the size of the training set. To cope with this, we cluster the training samples into K clusters and the nearest neighbour sample(s) are selected from the same cluster of X, and far-away samples are randomly selected from clusters that have faraway centers from the cluster to which X belongs. This

simple technique speeds up the training process drastically. Furthermore, instead of training $\mathcal{E}_\phi + \mathcal{D}$ from scratch, the wights of these networks are initialized based on an optimized traditional encoder-decoder network.

The hyperparameters λ_i have a key role on the final performance of the network and can be set dependent on application. After a joint training of the $\mathcal{E}_\phi$+$\mathcal{D}$, and with respect to the values of $\lambda_{i\in\{1,2,3\}}$, the networks can be interpreted as the following:

- $||X - \mathcal{D}(\mathcal{E}_\phi(X))||^2 < \epsilon_1, ||X_0 - \mathcal{D}(\mathcal{E}_\phi(X))||^2 < \epsilon_2$, where ϵ_1 and ϵ_2 are small non-negative scalars. But $||X_\infty - \mathcal{D}(\mathcal{E}_\phi(X))||^2 > \epsilon_3$, where ϵ_3 is very larger than ϵ_1 and ϵ_2. As aforementioned, X_0 and X_∞ are close to and far from X, respectively. Consequently, we can say that with a high probability X_0 and X come from the same class and X_∞ from another class. Accordingly, $\mathcal{D}(\mathcal{E}_\phi(X))$ is forced to be close to the samples from the same class and away from samples of other classes, leading to more separable samples in the reconstructed space.

- Let $\mathcal{P}_c$ be the probability of an specific classifier labeling X as class c. We expect that $\mathcal{P}_c(\mathcal{E}_\phi(X))|_{\frac{1}{3},\frac{1}{3},\frac{1}{3}} > \mathcal{P}_c(\mathcal{E}_\phi(X))|_{1,0,0}$. The subscripts $|_{\lambda_1,\lambda_2,\lambda_3}$ denote the values of λ_1, λ_2, and λ_3, respectively. This is because $\mathcal{E}_\phi$ is forced to map sample X to a latent-space with enough neighborhood information that will result in a more separable decoding.

- In a classification problem, if X belongs to the class c, and λ_2 and λ_3 are selected large enough, it is expected that $\mathcal{P}_c(D(\mathcal{E}_\phi(X + \sigma)))|_{\lambda_1,\lambda_2,\lambda_3} > \mathcal{P}_c(D(\mathcal{E}_\phi(X + \sigma)))|_{1,0,0}$, where σ denotes noise element. Our model considers the relation of sample X with its neighbors to make the model robust against noise and outliers. Similar concept is investigated in [19]. This characteristic of our relational reconstruction is an effective mechanism for defense against adversarial attacks. To defend deep networks against adversarial example attacks, reconstruction of adversarial examples using our encoder-decoder formulation trained on original (clean and normal) sample set is very useful. Similar argument can be found in recent defence mechanisms such as MagNet [35] and defense-GAN [53].

- $||X - \mathcal{D}(\mathcal{E}_\phi(X))|_{\lambda_1,\lambda_2\neq 0,\lambda_3\neq 0}||^2 \approx ||X - \mathcal{D}(\mathcal{E}_\phi(X))|_{1,0,0}||^2$. This implies that although our formulation (*i.e.*, learning to reconstruct an example with respect to neighborhood and relational information) does not just focus on the reconstruction, after training it is still able to efficiently reconstruct the input samples (see Fig. 4). Furthermore, our formulation does not over-emphasize on reconstruction loss only and borrows information from the neighborhood embedding. Therefore, it reconstructs data based on semantics instead of pixel value loss functions.

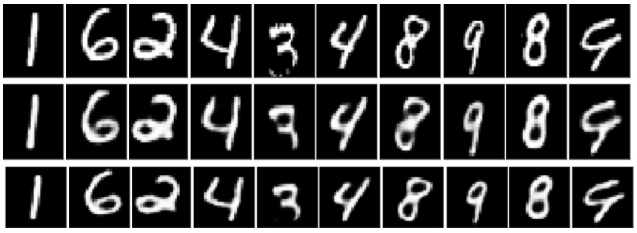

Figure 4. Several examples of reconstructed images from the original MNIST sample (1^{st} row) using a conventional AE (3^{rd} row) and our proposed encoder-decoder network (2^{nd} row) . Conventional network is optimized based on the reconstruction errors and our network is trained to optimize the reconstruction error alongside of neighborhood-relational information, Eq. (1).

We know that the relational information contains important cues, but paying extra attention to them and not properly preserving the context of samples (*i.e.*, the reconstruction error) may conclude adverse results. Normally, the relational information is exploited besides context, as a side-information. Accordingly, to create a trade-off between these two sources of information, we set $\lambda_1 > \lambda_2, \lambda_3$.

4. Experimental Results

In this section, the proposed method is evaluated on different datasets and tasks, to showcase its reliability and generality. The performance results are analyzed in details and are compared with state-of-the-art techniques. To show the adaptability and generality of the introduced framework for a wide range of applications it is evaluated as (1) an auto-encoder (R-AE), (2) a self-supervised (unsupervised) representation learning method, (3) as a defense approach against adversarial example attacks, and finally for (4) anomaly detection. Our results are at least comparable or better than state-of-the-art methods in each of these fields.

4.1. Setup

Several deep networks are exploited in our experiments, which are explained in details in the supplementary material[1]. The weights of network $\mathcal{E}_\phi + \mathcal{D}$ are initialized based on the Adam optimizer and learning rate is set to 0.0001. Depending on the task, λ_1, λ_2, and λ_3 are selected and are shown by a triplet $(\lambda_1, \lambda_2, \lambda_3)$, subscripted for each method. All the reported results in this section are from our implementation using Tensor-Flow framework [1] and Python ran on an NVIDIA TITAN X.

4.2. Unsupervised Learning with NRE

Conventional formulations of AE were widely used as a popular tool for unsupervised feature learning. In the re-

[1]More details at: https://github.com/Sabokrou/NRE

Table 1. Compassion results of the accuracy for our method (NRE) with conventional and widely-used auto-encoders. The best results are typeset in bold. NRE is subscripted with the chosen hyperparameters (*i.e.*, $\text{NRE}_{\lambda_1,\lambda_2,\lambda_3}$).

Classifer	L-SVM		R-SVM	
# of Epochs	40	100	40	100
AE [20]	0.969	0.969	0.961	0.972
DAE [57]	0.942	0.936	0.954	0.964
Context AE [42]	0.970	0.974	0.978	0.981
Split-Brain AE [64]	0.972	0.973	0.975	0.979
$\text{NRE}_{0.5,0.2,0.3}$ (Ours)	**0.977**	**0.978**	**0.981**	**0.984**

cent years, new versions of AE such as split-brain [64], adversarial auto-encoder [34], and context auto-encoder [42] were presented. We evaluate the performance of our method (NRE) on MNIST dataset [28] and compare the results with these various types of AE. MNIST[2] dataset includes 60,000 handwritten digits from '0' to '9', with 50,000 and 10,000 samples as training and testing data, respectively.

Results on MNIST To evaluate the performance of various version of AEs, auto-encoders are trained with respect to different policies (objective functions). After training these networks, all training and testing samples are mapped to the AE latent representation space based on trained auto-encoders. Equally for each AE, a Support Vector Machine (SVM) [56] classifier on top of the represented training samples is trained and the classification accuracy on test set is reported. The results are shown in Table 1. As can be seen, the classification accuracy based on relational encoding is better than other methods. For fair comparison, all AEs are trained for 40 and 100 epochs. The classification is done by both a Linear SVM (L-SVM) and one with an RBF kernel (R-SVM). The hyperparameter of RBF is set as 0.01 and is fixed in all experiments.

Our proposed objective function is a more complex one compared to the conventional AE, and therefore it will achieve better results when it is trained for more epochs (even more than 100 epochs). But for fairness, all methods are trained for the same number of epochs.

4.3. Classification, Detection, and Segmentation

As mentioned before, unsupervised or self-supervised representation is increasingly used for different applications because of its advantage of not requiring labeled data. A pretext task is often first trained to direct the ultimate network to have proper initialization or even create the embedding space for the subsequent task. We compare our approach to the state-of-the-art methods, which all use variants of AlexNet [26]. We follow [64], for evaluating and comparing our method with the others. We pre-trained our network to learn the relational information on the ImageNet dataset [11]. This dataset is very large, so finding X_0 and X_∞ is very expensive and time-consuming. To end this, as mentioned in Section 3.2, the dataset is divided to $K = 400$ clusters and then just the partition involving any specific X is searched for finding its X_0 and X_∞ is randomly selected from clusters with faraway centers from the cluster of X. We evaluate the performance of our relational representation on PASCAL VOC dataset [15] as a benchmark set for classification tasks. This classication task involves 20 binary classication decisions regarding the presence or absence of 20 object classes. We used the AlexNet architecture and embeded it as the decoder in our AE formulation. We mirrored same architecture for the decoder by converting the convolutional and sub-sampling layers to deconvolutional and up-sampling layers.

[2] Available at http://yann.lecun.com/exdb/mnist/

Table 2. Performance of our self-supervised representation based on NRE for classification, detection, and segmentation tasks. Classification and Fast R-CNN [17] detection results for the PASCAL VOC 2007 [14] test set, and FCN [32] segmentation results on the PASCAL VOC 2012 validation set, under the standard mean average precision (mAP) or mean intersection over union (mIOU) metrics for each task. Classification, Det and Seg columns show classification, detection, and segmentation results, respectively.

	Classification			Det	Seg
Layers	FC8	FC6-8	all	all	all
AlexNet [26]	77.0	78.8	78.3	56.8	48.0
Agrawal *et al.* [2]	31.2	31.0	54.2	43.9	–
Pathak *et al.* [42]	30.5	34.6	56.5	44.5	30.0
Wang *et al.* [59]	28.4	55.6	63.1	47.4	–
Doersch *et al.* [12]	44.7	55.1	65.3	51.1	–
K-means [25]	32.0	39.2	56.6	45.6	32.6
AE [20]	24.8	16.0	53.8	41.9	–
BiGAN [13]	41.7	52.5	60.3	46.9	35.2
Counting [38]	–	–	67.7	51.4	36.6
Owens *et al.* [39]	–	–	61.3	44.0	–
Pathak *et al.* [42]	–	–	61.0	52.2	–
Jenni *et al.* [22]	–	–	69.8	52.5	38.1
DeepCluster [7]	–	–	73.7	55.4	45.1
Noorozi & Favaro [37]	–	–	67.6	53.2	37.6
$\text{NRE}_{0.5,0.25,0.25}$ (Ours)	**55.9**	**71.2**	**74.4**	**54.7**	**51.1**

Results on PASCAL VOC Several classifiers are trained by freezing various parts of the AlexNet [26]. In the first experiment, on top of FC6 and FC7, a linear classifier is trained. In the second experiment, all three FC6, FC7 and FC8 layers are trained in a supervised manner, where all other layers were frozen. Finally, the entire network is 'fine-tuned'. Table 2 compares our results in comparison with the state-of-the-art methods.

We further evaluated object detection and segmentation tasks with the pre-trained AlexNet used as the initializa-

Figure 5. Some samples of adversarial examples created by FGSM [40] attack ($\epsilon = 0.2$). First row: Original images; Second row: reconstruction of the adversarial examples by our method; Last row: the adversarial example.

tion for Fast R-CNN [17] and fully convolutional network (FCN) [32], object detection and segmentation tasks, respectively. For these tests, we replaced the supervised trained AlexNet [26] with our self-supervised trained network, as a pre-training for the specific task. Results confirm that the proposed method can be used as an efficient approach for self-supervised feature learning. In all cases (except for the segmentation task) our results are superior to others by a considerable margin.

4.4. Defense Against Adversarial Attacks

Adversarial examples are means of fooling trained networks for specific computer vision and are a challenging problem with respect to the safety of deep networks. Let $\mathcal{F}$ be a classifier, which has correctly labeled X as Y, *i.e.*, $\mathcal{F}(X) = Y$. Adversarial attack is done by contaminating X in a way that leads to creating its equivalent adversarial example $\hat{X}$, where $||X - \hat{X}|| < \epsilon$ (ϵ is a small non-negative scalar) and $\mathcal{F}(\hat{X}) \neq Y$. This defines a vulnerability for the classifier $\mathcal{F}$ [35, 53].

As a defense mechanism, MagNet [35] has proposed to refine the adversarial example using an auto-encoder using the manifold distribution of the correct class. Here, we show that our proposed encoder-decoder with NRE performs better than MagNet. We evaluate our method and MagNet [35] with respect to Fast Gradient Sign Attack (FGSM) [40] attack with different values of ϵ. These experiments are done on the MNIST dataset. We select 50,000 examples for the training set, 150 samples for training the substitute network, and finally 9850 samples for testing. In a black-box attack, the attacker does not have access to the architecture and weights of target classifier. But it is possible to emulate the behaviour of target classifier using a substitute network, learned on 150 samples. We trained a CNN network as our target classifier and obtained an accuracy of 98.6%, alongside this classifier, a substitute CNN classifier is trained on 150 samples with a 77% accuracy.

Table 3. Performance evaluation of NRE for refinement of adversarial examples as a defense strategy against adversarial attacks made by the black-box FGSM [40] attack. Best results in each column are typeset in bold.

ϵ	0.01	0.1	0.2	0.3
MagNet [35]	0.9238	0.7655	0.614	0.4242
$\text{NRE}_{0.6,0.2,0.2}$ (Ours)	**0.9802**	**0.9056**	**0.8489**	**0.7166**

Results of Defense Against Attacks We evaluate the performance of NRE as a defense approach against the adversarial examples and report the results in Table 3. This table shows our results in comparison to MagNet [35] as a baseline, which is based on reconstructing the adversarial sample. The architectures used for the auto-encoders of both methods are the same, but are learned with different object functions. A FGSM attack with different amounts of ϵ is used. Several adversarial examples and their reconstructions by NRE are shown in Fig. 5. Naturally, after the attacks the accuracy of network is decreased. Our method and MagNet are applied on adversarial examples to refine them as a pre-processing step for the classifier. It can be seen that the accuracy of our method for all values of ϵ is better than MagNet by a wide margin.

There are more types of attacks and defense strategies against adversarial attacks, such DefenseGAN [53]. Analyzing all of these attacks and defense methods requires a deep discussion, which is out of the scope of this paper. Here, we briefly compared our method to the state-of-the-art to showcase efficiency of our method.

4.5. Video Anomaly Detection

Detecting Anomalous events in videos (also refered to as irregularity detection in visual data) is an important task in different computer vision application. Recent state-of-the-art methods for anomaly detection are often based on encoder-decoder networks and analyzing the reconstruction error. As the context of our network is very close to these solutions for anomaly detection, we evaluate our method on this task as well.

We evaluate our method on the UCSD Ped2 dataset [8], which is a popular dataset for this task. We follow the evaluation criteria of [51]. Similar to [51], the frame-level accuracy is reported as the performance metric. In frame-level measure, a frame is considered as anomaly, if at least one of its pixels is detected as anomaly. The UCSD dataset has two subsets, referred to as Ped1 and Ped2. They are from different static-camera outdoor scenes, with 10 fps and resolutions of 158×234 and 240×360, respectively. Moving objects in these videos are mainly pedestrians, and all other objects like cars, wheelchairs, and bicycles are labeled as anomaly. To compare with the previous work on this dataset, we evaluate our algorithm on Ped2.

Table 4. Frame-level anomaly detection comparisons on UCSD Ped2 in terms of Equal Error Rates (EER).

Method	EER	Method	EER
IBC [5]	13%	RE [46]	15%
MPCCA [24]	30%	Ravanbakhsh *et al.* [45]	13%
MDT [33]	24%	Ravanbakhsh *et al.* [44]	14%
Bertini *et al.* [4]	30%	Dan Xu *et al.* [62]	17%
Deep-Anomaly[50]	13.5%	Sabokrou *et al.* [47]	19%
Li *et al.* [31]	18.5%	Deep-cascade[49]	**9%**
AVID [52]	14%	ALOCC[51]	13%
$\text{NRE}_{0.6,0.2,0.2}$	17.5%	$\text{NRE}_{0.6,0.4,0}$	14%

Anomaly Detection Results For this experiment, we divided the video frames into 2D patches of size 30×30. All patches extracted from normal frames are considered for our training. Note that training data only contains normal patches. An encoder-decoder network with our objective function on all training patches is trained (See subsection 3.2). When the training is completed, test patches are fed to this encoder-decode network one by one. Similar to [46], regular reconstruction error ($||X - \hat{X}||^2$) is used as a measure for detecting the anomalies. If this reconstruction error is larger than a threshold, it means that the patch contains something that was not seen during training (*i.e.*, it is an anomaly). Our method is very similar to [46], but with two major differences: (1) We use only one auto-encoder but [46] has exploited two auto-encoder; (2) Our auto-encoder is learned based on relational-information, while [46] only trained based on reconstruction error.

Table 4 reports the results of our method and other baseline and state-of-the-art approaches. Last row shows the results of our method with two different values for λ_1, λ_2, and λ_3. As can be seen, our method is comparable with the state-of-the-art. NRE is a very simple method based on the sole criteria of reconstruction error and neighborhood relational encoding, while other methods (such as [47] and [49]) are based on intensive spatio-temporal embedding of video content. The experiments show the generality of the proposed approach. We report our results with respect to different values of λ_1, λ_2, λ_3, and $T = 1$.

5. Discussion

The results confirm that the proposed neighborhood relational encoding method for learning unsupervised and self-supervised representation can be adapted for a variety of computer vision and image analysis tasks. There are several challenges and interesting intuitions around NRE, which are discussed in the following:

Values of λ: The objective function consists of three terms, which can be adjusted depending on the target task. Our results show that λ_1 is very important on all types of tasks. But for tasks such as de-nosing, in-painting, and generally enhancing the images, λ_1 and λ_2 show to be more important than λ_3. For classification and clustering tasks that require creating discriminative embedding spaces equal values of λ_1 and $\lambda_2 + \lambda_3$ often result in better performance.

The hyperparameter T: T is a very important hyperparameter for capturing the intrinsic neighborhood-relational-information. Obviously, the larger it is selected to be, the more robustness is added for the method against outliers. However, if it is set to be very large, the concept of neighborhood will be lost. Therefore, a compromise should be made for each specific task.

Dynamic λs: Scheduling the values of λ_1, λ_2, and λ_3 during training can be very useful and lead to speed-ups in convergence. This can be a very interesting direction for the future work, as designing a good scheduling for gradually changing of these parameters (while interacting with each other) is not a straightforward task.

Metrics for finding similar or dissimilar images: The main difficulty of the proposed method is finding the similar and dissimilar samples to any target image. We tested a wide range of metrics and found that simple cosine similarity in the latent space was enough for this purpose. Comparing images in their original space (pixel values) is not an appropriate option here, as it neglects the important context and semantics embedded in images. Better metrics can improve the results and be developed for specific applications.

6. Conclusion

In this paper, we proposed a learning framework for encoder-decoder networks (*e.g.*, auto-encoders), and adopted it for a wide range of computer vision tasks. Our proposed method encodes the neighborhood-relations information into the AE and turns it to a kernel embedding framework. Therefore, besides learning a reconstruction scheme, our AE preserves the local geometric manifold. This leads to dicriminative neighborhood-guided self-supervised representation learning that can be used in a variety of applications, since it does not require label information for training. We evaluated our models in different related applications including self-supervise (unsupervised) representation learning for classification, detection, and segmentation, as well as defense against adversarial example attacks and anomaly detection in videos. The results suggest that our method is superior to, or at least comparable with, the state-of-the-art specific to each application while being much simpler.

Acknowledgement M. Sabokrou was in part supported by a grant from IPM (No. CS1396-5-01). E. Adeli would like to thank Panasonic for the support.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016. 5

[2] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *ICCV*, pages 37–45, 2015. 2, 6

[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 1

[4] Marco Bertini, Alberto Del Bimbo, and Lorenzo Seidenari. Multi-scale and real-time non-parametric approach for anomaly detection and localization. *Computer Vision and Image Understanding*, 116(3):320–329, 2012. 8

[5] Oren Boiman and Michal Irani. Detecting irregularities in images and in video. *International journal of computer vision*, 74(1):17–31, 2007. 8

[6] Biagio Brattoli, Uta Büchler, Anna-Sophia Wahl, Martin E Schwab, and Björn Ommer. Lstm self-supervision for detailed behavior analysis. In *CVPR*, volume 2, 2017. 2

[7] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018. 1, 6

[8] Antoni Chan and Nuno Vasconcelos. Ucsd pedestrian dataset. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(5):909–926, 2008. 7

[9] Wenqing Chu and Deng Cai. Stacked similarity-aware autoencoders. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1561–1567. AAAI Press, 2017. 2

[10] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967. 3

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 6

[12] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *CVPR*, pages 1422–1430, 2015. 2, 6

[13] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016. 6

[14] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015. 6

[15] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 6

[16] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 1

[17] Ross Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015. 6, 7

[18] Jihun Ham, Daniel D Lee, Sebastian Mika, and Bernhard Schölkopf. A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the twenty-first international conference on Machine learning*, page 47, 2004. 2

[19] Xiaofei He, Deng Cai, Shuicheng Yan, and Hong-Jiang Zhang. Neighborhood preserving embedding. In *ICCV*, volume 2, pages 1208–1213. IEEE, 2005. 5

[20] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006. 2, 6

[21] Dinesh Jayaraman and Kristen Grauman. Learning image representations tied to ego-motion. In *ICCV*, pages 1413–1421, 2015. 2

[22] Simon Jenni and Paolo Favaro. Self-supervised feature learning by learning to spot artifacts. In *CVPR*, pages 2733–2742, 2018. 6

[23] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014. 1

[24] Jaechul Kim and Kristen Grauman. Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. In *CVPR*, pages 2921–2928, 2009. 8

[25] Philipp Krähenbühl, Carl Doersch, Jeff Donahue, and Trevor Darrell. Data-dependent initializations of convolutional neural networks. *arXiv preprint arXiv:1511.06856*, 2015. 6

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1, 2, 6, 7

[27] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *ECCV*, pages 577–593. Springer, 2016. 2

[28] Yann LeCun, Corinna Cortes, and Christopher Burges. Mnist dataset. *URL http://yann. lecun. com/exdb/mnist*, 1998. 6

[29] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *ICCV*, pages 667–676. IEEE, 2017. 1

[30] Minxian Li, Xiatian Zhu, and Shaogang Gong. Unsupervised person re-identification by deep learning tracklet association. In *ECCV*, 2018. 1

[31] Weixin Li, Vijay Mahadevan, and Nuno Vasconcelos. Anomaly detection and localization in crowded scenes. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):18–32, 2014. 8

[32] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 6, 7

[33] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. Anomaly detection in crowded scenes. In *CVPR*, pages 1975–1981, 2010. 8

[34] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015. 1, 2, 6

[35] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147. ACM, 2017. 1, 2, 5, 7

[36] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, pages 527–544. Springer, 2016. 2

[37] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016. 1, 2, 6

[38] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. 1, 2, 6

[39] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. Ambient sound provides supervision for visual learning. In *ECCV*, pages 801–816. Springer, 2016. 2, 6

[40] Nicolas Papernot, Nicholas Carlini, Ian Goodfellow, Reuben Feinman, Fartash Faghri, Alexander Matyasko, Karen Hambardzumyan, Yi-Lin Juang, Alexey Kurakin, Ryan Sheatsley, et al. cleverhans v2. 0.0: an adversarial machine learning library. *preprint arXiv:1610.00768*, 2016. 7

[41] Deepak Pathak, Ross B Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *CVPR*, volume 1, page 7, 2017. 2

[42] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 2, 6

[43] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, pages 2536–2544, 2016. 1, 2

[44] Mahdyar Ravanbakhsh, Moin Nabi, Enver Sangineto, Lucio Marcenaro, Carlo Regazzoni, and Nicu Sebe. Abnormal event detection in videos using generative adversarial nets. In *IEEE International Conference on Image Processing (ICIP)*, 2017. 8

[45] Mahdyar Ravanbakhsh, Enver Sangineto, Moin Nabi, and Nicu Sebe. Training adversarial discriminators for cross-channel abnormal event detection in crowds. *arXiv preprint arXiv:1706.07680*, 2017. 8

[46] M Sabokrou, M Fathy, and M Hoseini. Video anomaly detection and localisation based on the sparsity and reconstruction error of auto-encoder. *Electronics Letters*, 52(13):1122–1124. 2, 8

[47] Mohammad Sabokrou, Mahmood Fathy, Mojtaba Hoseini, and Reinhard Klette. Real-time anomaly detection and localization in crowded scenes. In *CVPR Workshops*, pages 56–62, 2015. 8

[48] Mohammad Sabokrou, Mahmood Fathy, Zahra Moayed, and Reinhard Klette. Fast and accurate detection and localization of abnormal behavior in crowded scenes. *Machine Vision and Applications*, 28(8):965–985, 2017. 1

[49] Mohammad Sabokrou, Mohsen Fayyaz, Mahmood Fathy, and Reinhard Klette. Deep-cascade: Cascading 3d deep neural networks for fast anomaly detection and localization in crowded scenes. *IEEE Transactions on Image Processing*, 26(4):1992–2004, 2017. 8

[50] Mohammad Sabokrou, Mohsen Fayyaz, Mahmood Fathy, Zahra Moayed, and Reinhard Klette. Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes. *Computer Vision and Image Understanding*, 172:88–97, 2018. 8

[51] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. Adversarially learned one-class classifier for novelty detection. In *CVPR*, pages 3379–3388, 2018. 2, 3, 7, 8

[52] Mohammad Sabokrou, Masoud Pourreza, Mohsen Fayyaz, Rahim Entezari, Mahmood Fathy, Jürgen Gall, and Ehsan Adeli. Avid: Adversarial visual irregularity detection. *ACCV*, 2018. 1, 2, 8

[53] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *ICLR*, 2018. 2, 5, 7

[54] Sasan Sharifipour, Hossein Fayyazi, Mohammad Sabokrou, and Ehsan Adeli. Unsupervised feature ranking and selection based on autoencoders. In *ICASSP*, pages 3172–3176. IEEE, 2019. 1

[55] John Shawe-Taylor, Nello Cristianini, et al. *Kernel methods for pattern analysis*. Cambridge university press, 2004. 3

[56] Vladimir Vapnik, Isabel Guyon, and Trevor Hastie. Support vector machines. *Mach. Learn*, 20(3):273–297, 1995. 6

[57] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008. 1, 2, 6

[58] Shuyang Wang, Zhengming Ding, and Yun Fu. Feature selection guided auto-encoder. In *AAAI*, pages 2725–2731, 2017. 1

[59] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, pages 2794–2802, 2015. 1, 2, 6

[60] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 4

[61] Zhirong Wu, Yuanjun Xiong, X Yu Stella, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742, 2018. 1

[62] Dan Xu, Elisa Ricci, Yan Yan, Jingkuan Song, and Nicu Sebe. Learning deep representations of appearance and motion for anomalous event detection. In *BMVC*, 2015. 8

[63] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, pages 649–666. Springer, 2016. 2

[64] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, volume 1, page 5, 2017. 1, 2, 6

[65] Qingyu Zhao, Nicolas Honnorat, Ehsan Adeli, Adolf Pfefferbaum, Edith V Sullivan, and Kilian M Pohl. Variational autoencoder with truncated mixture of gaussians for functional connectivity analysis. In *International Conference on Information Processing in Medical Imaging*, pages 867–879. Springer, 2019. 1