Auto-encoder factorizing into transform invariants and transform parameters*

Tadashi Matsuo
College of Information Science and Engineering
Ritsumeikan University
Shiga, Japan
matsuo@i.ci.ritsumei.ac.jp

Nobutaka Shimada
College of Information Science and Engineering
Ritsumeikan University
Shiga, Japan
shimada@ci.ritsumei.ac.jp

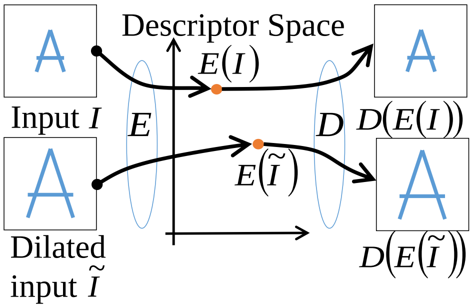

Fig. 1. Characteristics of an ordinary auto-encoder

***Abstract*—The auto-encoder method is a type of unsupervised dimensionality reduction method. However, it is difficult to use an ordinary auto-encoder for encoding a spatial pattern itself because such ordinary one encodes an image and its spatially dilated/reduced versions into descriptors far from each other. This will be a problem when focusing on a pattern itself. To solve this, we proposed a transform invariant auto-encoder based on a cost function evaluating transform invariance. By the method, we can extract a transform invariant descriptor from an input, but we need an additional regressor to extract transform parameters required to restore the input. In addition, the cost function requires high computation cost by computational explosion when considering multiple types of transforms.**

In this publication, we propose a novel auto-encoder that separates an input into a transform invariant descriptor and transform parameters. The proposed method does not require an additional regressor and it will overcome combinational explosion. The proposed method can be applied to various auto-encoders without requiring any special modules or labeled training samples. By applying it to dilation transforms, we can achieve a spatial pattern descriptor and its relative scale. By some experiments, we demonstrate that the method can generate a pair of a transform invariant descriptor and a set of parameters for restoring the original input.

***Index Terms*—auto-encoder, unsupervised learning, machine learning**

I. Introduction

The auto-encoder method [1]–[3] is a type of dimensionality reduction method. It can extract essential information from a vector via general non-linear mapping. Moreover, a mapping from a vector to a descriptor representing essential information can be automatically generated from a set of vectors without any supervising information.

When encoding images by the auto-encoder method, a descriptor of an image generally differs from that of a spatially dilated version of the image as shown in Fig. 1, because a pattern itself and its scale are inseparably embedded into a descriptor. Although the denoising auto-encoder method [4] can extract desired components from an input including information to be ignored, it requires an ideal output for each training sample when training an auto-encoder. Therefore, to generate a descriptor representing a spatial pattern in an image by such an auto-encoder, we need to normalize its scale in the images prior to training the auto-encoder. However, such a spatial normalization is generally difficult. For example, the normalization of the appearances of various hand–object interactions is not obvious and requires a pattern recognition technique to automatically find the standard for each image.

We have proposed a transform invariant auto-encoder that generates a descriptor invariant with respect to a set of transforms [5]. By considering spatial dilations, the method can generate a dilation invariant auto-encoder, which extracts a typical spatial pattern without regard to its relative scale (Fig. 2). Since the framework does not depend on a structure of an auto-encoder, it can be applied to various auto-encoders without requiring any special modules or labeled training samples. By using the method, we can encode a spatial pattern itself even if target images are difficult to label or normalize, for example, the appearances of hand–object interactions. However, it ignores a scale of the pattern. To estimate the scale of the pattern, we had to introduce an additional inference model.

In this paper, we propose a novel auto-encoder that separates an input into a transform invariant descriptor and transform parameters. It consists of a transform invariant encoder, the corresponding decoder and a regressor of transform parameter as shown in 3. The encoder, decoder and regressor can be trained simultaneously and an external additional regressor is not required. The proposed method can be applied to various auto-encoders without requiring any special modules or labeled training samples. In addition, the proposed method will overcome combinational explosion, which occurs

*This work was supported by JSPS KAKENHI Grant Number 18H03313.

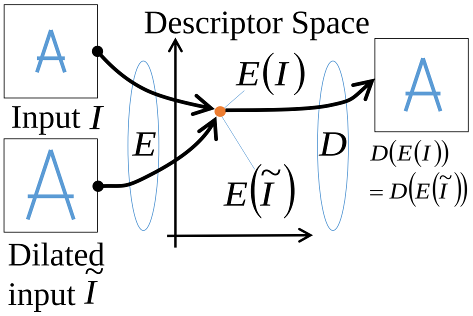

Fig. 2. Characteristics of a dilation invariant auto-encoder

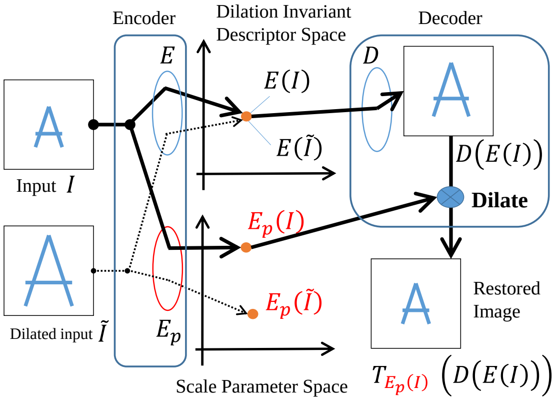

Fig. 3. Characteristics of a proposed auto-encoder

a problem when training a transform invariant auto-encoder for very widely various transforms. By applying it to dilation transforms, we can achieve a spatial pattern descriptor and its relative scale. By an experiment, we demonstrate that the method can generate a pair of a transform invariant descriptor and a set of parameters for restoring the original input.

II. Ordinary auto-encoder

In general, an auto-encoder is so trained that the encoder–decoder combination approximately restores an input in a certain input set. It is formulated as a problem minimizing a cost function $C_{\text{ord}}(E, D)$ defined as

$$C_{\text{ord}}\left(E, D\right) = \sum_{I \in S} \left\| I - D\left(E\left(I\right)\right) \right\|_2^2 , \tag{1}$$

where S, $E(\cdot)$, $D(\cdot)$, and $\left\|\cdot\right\|_p$ denote a set of inputs, the encoder, the decoder, and the ℓ^p norm, respectively.

To minimize $C_{\text{ord}}(E, D)$, the decoder should be able to approximately restore an original vector I from its descriptor $E(I)$, which has a lower dimensionality than I. By training the encoder E and the decoder D by minimizing $C_{\text{ord}}(E, D)$, information sufficient to restore an original vector can be extracted as a descriptor by the encoder. In this way, the auto-encoder method can construct descriptors of vectors from just a set of training vectors.

However, a descriptor of an image from an ordinary auto-encoder includes both a spatial pattern and its position. If images have a common spatial pattern at different positions, their descriptors are different.

III. Transform invariant auto-encoder

We have proposed the transform invariant auto-encoder method [5]. It is trained by minimizing the following cost function;

$$\begin{aligned} C_{\text{old}}\left(E, D\right) = \sum_{I \in S} & \lambda_{\text{inv}} \sum_i \left\| D\left(E\left(I\right)\right) - D\left(E\left(T_{\theta_i}\left(I\right)\right)\right) \right\|_2^2 \\ & + \lambda_{\text{res}} \min_\theta \left\| I - T_\theta\left(D\left(E\left(I\right)\right)\right) \right\|_2^2 . \\ & + \lambda_{\text{spa}} \left(\frac{\left\| E\left(I\right) \right\|_1}{\left\| E\left(I\right) \right\|_2} \right)^2 , \end{aligned} \tag{2}$$

where S and T_θ denote a set of training inputs and a transform operator in the ignored transforms, respectively.

By minimizing the above cost function, we can achieve an auto-encoder that is transform invariant and can restore a pattern accurately. However, calculation of the function may require high computation cost for various transforms because the function includes minimization with respect to the transform parameter θ.

IV. Proposed method

We propose a new auto-encoder that separates an input into a transform invariant descriptor and transform parameters. The basic idea is relaxation of the minimization of the restoration term (the third term in (1)) for the transform invariant auto-encoder. To calculate the restoration term, it is required to find the transform parameter θ giving the minimum. However, it is generally difficult when a transform parameter is continuous and high-dimensional. So, we propose a method to avoid searching the concrete minimum on the whole transform parameter space by using a weight function, which indicates a transform parameter near to the minimum. The weight function can be used as a regressor of a transform parameter for an input. The weight function can be optimized simultaneously with the transform invariant encoder and the corresponding decoder.

A. Cost function

Searching the minimum can be replaced with optimization of the weight function $W(\theta)$ as follows;

$$\min_{\theta \in \Theta} f(\theta) = \min_{W(\theta) \geq 0, \int_\Theta W(\theta) d\theta = 1} \int_\Theta f(\theta) W(\theta) d\theta, \tag{3}$$

where

$$f(\theta) \overset{\text{def}}{=} \left\| I - T_\theta\left(D\left(E\left(I\right)\right)\right) \right\|_2^2 . \tag{4}$$

If the integral in the right side of (3) is near to the minimum, the weight function $W(\theta)$ will be large on a small neighborhood of the minimum and almost zero otherwise. This means that the weight function indicates the parameter giving the minimum. Moreover, the weight function W can be designed so that it can be optimized by gradient method even if it is difficult to differentiate $f(\theta)$ itself. In addition, the weight function $W(\theta)$ may depend on each input I. Therefore, the function W can be minimized simultaneously with the transform invariant encoder E and the corresponding decoder D.

By considering continuous parameters, we can rewrite the cost function for the transform invariant auto-encoder as following;

$$\begin{aligned}&\sum_{I\in S}\lambda_{\text{inv}}\int_{\Theta}\left\|D\left(E\left(I\right)\right)-D\left(E\left(T_{\theta}\left(I\right)\right)\right)\right\|_2^2 d\theta\\&\quad+\lambda_{\text{res}}\min_{W}\int_{\Theta}\left\|I-T_{\theta}\left(D\left(E\left(I\right)\right)\right)\right\|_2^2 W(I,\theta)d\theta\\&\quad+\lambda_{\text{spa}}\left(\frac{\left\|E\left(I\right)\right\|_1}{\left\|E\left(I\right)\right\|_2}\right)^2,\end{aligned}\tag{5}$$

where W is optimized under the condition that $0\leq W(I,\theta)$ and $\int_{\Theta}W(\theta)d\theta=1$. By optimizing W for the total value instead for only the restoration term, we can define a new cost function as following;

$$\begin{aligned}&C\left(E,D,W\right)\overset{\text{def}}{=}\\&\sum_{I\in S}\lambda_{\text{inv}}\int_{\Theta}\left\|D\left(E\left(I\right)\right)-D\left(E\left(T_{\theta}\left(I\right)\right)\right)\right\|_2^2 d\theta\\&+\lambda_{\text{res}}\int_{\Theta}\left\|I-T_{\theta}\left(D\left(E\left(I\right)\right)\right)\right\|_2^2 W(I,\theta)d\theta\\&+\lambda_{\text{spa}}\left(\frac{\left\|E\left(I\right)\right\|_1}{\left\|E\left(I\right)\right\|_2}\right)^2.\end{aligned}\tag{6}$$

We train the transform invariant encoder E, the corresponding decoder D and the transform parameter weight function W so that they minimize the proposed cost function $C\left(E,D,W\right)$.

B. Calculation

For convenience of calculation, we suppose that the transform parameter weight function $W(I,\theta)$ is a Gaussian function on the transform parameter space as follows;

$$W\left(I,\theta\right)=\frac{1}{\left(2\pi\right)^{\frac{D}{2}}\left|\frac{1}{2}\Sigma_I\right|^{\frac{1}{2}}}e^{-\frac{1}{2}\left(\theta-\mu_I\right)^T\left(\frac{1}{2}\Sigma_I\right)^{-1}\left(\theta-\mu_I\right)},\tag{7}$$

where D denotes the dimension of a transform parameter and Σ and μ denotes the scaled covariance matrix and the mean, respectively.

We use the Monte Carlo method to calculate integrals in the cost function (6). First, we define a utility function $w(I,\theta)$ as follows;

$$\begin{aligned}w\left(I,\theta\right)&\overset{\text{def}}{=}e^{-\frac{1}{2}\left(\theta-\mu_I\right)^T\Sigma_I^{-1}\left(\theta-\mu_I\right)},\\p_I\left(\theta\right)&\overset{\text{def}}{=}\frac{1}{\left(2\pi\right)^{\frac{D}{2}}\left|\Sigma_I\right|^{\frac{1}{2}}}e^{-\frac{1}{2}\left(\theta-\mu_I\right)^T\Sigma_I^{-1}\left(\theta-\mu_I\right)},\end{aligned}\tag{8}$$

where p means a probability density function of a Gaussian distribution. $W\left(I,\theta\right)$ can be represented as

$$W\left(I,\theta\right)=2^{\frac{D}{2}}w\left(I,\theta\right)p_I\left(\theta\right).\tag{9}$$

By using the utility functions, we can approximately calculate the restoration term, the second term in (6) as follows;

$$\begin{aligned}&\int_{\Theta}\left\|D\left(E\left(I\right)\right)-T_{\theta}\left(I\right)\right\|_2^2 W\left(I,\theta\right)d\theta\\&=2^{\frac{D}{2}}\int_{\Theta}\left\|I-T_{\theta}\left(D\left(E\left(I\right)\right)\right)\right\|_2^2 w\left(I,\theta\right)p_I\left(\theta\right)d\theta\\&\approx\frac{2^{\frac{D}{2}}}{N}\sum_{\theta_n\sim N_D\left(\mu_I,\Sigma_I\right)}\left\|I-T_{\theta_n}\left(D\left(E\left(I\right)\right)\right)\right\|_2^2 w\left(I,\theta_n\right),\end{aligned}\tag{10}$$

where $N_D\left(\mu_I,\Sigma_I\right)$ denotes the D-dimensional Gaussian distribution and N denotes the number of sampled parameters $\{\theta_n\}$. We minimize the cost function (6) by optimizing the parameters μ_I and Σ_I as functions of an input I. By using the Monte Carlo method, we can avoid combinational explosion when searching the minimum from whole possible transform parameters. Similarly, we can calculate the invariance term, the first term in (6), by the Monte Carlo method with the uniform distribution of possible transform parameters.

V. Experiments

We demonstrate the effectiveness of the proposed method by experiments with a dilation invariant auto-encoder. On the experiments, we supposed that a transform parameter θ consisted of a logarithmic scale θ_s and the dilation operator T_θ was defined as

$$\left(T_{\theta}\left(I\right)\right)\left(x,y\right)=I\left(\frac{x}{e^{-\theta_s}},\frac{y}{e^{-\theta_s}}\right),\tag{11}$$

where $I(x,y)$ denotes the value of the image I at the position (x,y). As a range of scales, we supposed that $0.5\leq e^{\theta_s}\leq 2$.

As a transform invariant encoder, we used a neural network consisting of a single CNN with 9×9 filter kernels and 16-channel outputs following a max pooling with stride 2 and a three-layer fully connected neural network (NN), where each layer has 1500, 150, 30 outputs respectively. As a decoder corresponding to the encoder, we used a three-layer fully connected NN, where each layer has 150, 1500, 1024 outputs, respectively. As a regressor of μ_I and Σ_I, which are parameters of a transform parameter weight function, we used a four-layer fully connected NN, where each layer has 256, 64, 16 and 2 outputs, respectively. The one of outputs is used as one dimensional μ_I and the other one is used for generating one dimensional Σ_I. In addition, we used a hyperbolic tangent as an activation function, which is placed between each pair of layers.

Here, we demonstrate dilation invariant property of the proposed method using experiments for digit patterns.

We trained an auto-encoder by minimizing (6) for digit images generated by randomly dilating training images in the MNIST database [6]. The auto-encoder was trained by stochastic gradient descent (SGD) [6] with learning rate 1.0×10^{-4}, and updated with every 10 samples that were randomly extracted from the training images in the MNIST database. We used the auto-encoder that were updated 10,000 times. Training the proposed auto-encoder took a little less than 33 hours.

If we use the transform invariant auto-encoder trained by minimizing (2), we need to discretize the tranform parameter space beforehand because the parameter is essentially continuous. With the proposed method, we do not have to design a discretized parameter space beforehand because transformed parameters are randomly sampled subject to the weight function when calculating the cost function (10).

As an example, we encoded and decoded training images of some digits, which were used in training the auto-encoder. The results are shown in Fig. 4, 5, 6 and 7. The first row of each figure consists of dilated input images, where the dilation ratios for each column are 0.5, 0.8, 1.1, 1.4, 1.7 and 2.0 (1.0 means the original scale in the MNIST database). The second row consists of images generated by encoding an image, which is placed on the corresponding column of the first row, and decoding it. The third row consists of images generated by dilating images in the second row with a mean transform parameter μ_I estimated from a corresponding input image. From the images in the second rows, decoded images for each digit have almost similar shape to input images and they shares an almost same scale even though input images have various scales. This means that the proposed auto-encoder successfully generated descriptors invariant for dilation transforms. In addition, the images in the third rows have a scale almost similar to each corresponding input image. This means that the proposed auto-encoder successfully extracted transform parameters from training images.

We also applied the auto-encoder to test images which were not used in training the auto-encoder. The results are shown in Fig. 8, 9, 10 and 11. Similarly to the results from training images, images in each second row shares an almost same scale and original scales are approximately restored in the corresponding third row. This means that the proposed auto-encoder successfully extracted transform parameters from test images.

VI. Conclusion

We proposed a novel auto-encoder that can separate an input into a transform invariant descriptor and transform parameters. By utilizing a transform parameter weight function and the Monte Carlo method, we can apply it to transforms with continuous parameters. And also we can avoid a problem of combinational explosion when training a proposed auto-encoder for various transforms. By an experiment, we showed that the auto-encoder can encode a pattern independently of its scale.

The framework of the proposed cost function can be applied to temporal patterns and more various transforms such as combination of spatial shifts, dilations and rotations. By using the proposed auto-encoder, we can extract typical patterns from complicated images, which does not have an obvious normalization, such as hand-object interactions or motions. This will be useful for interaction-based or motion-based recognition.

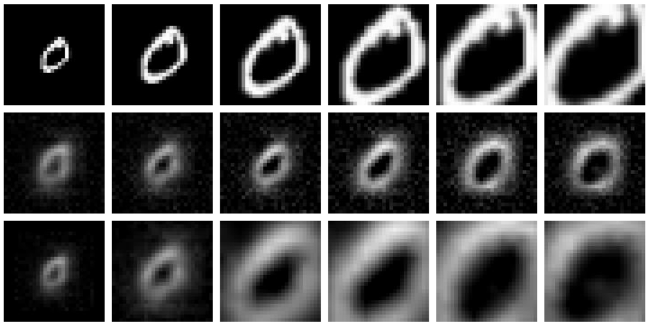

Fig. 4. Results for a training image of “0”: Dilated input training images (first row), images decoded from descriptors (second row) and decoded images dilated with estimated scales (third row)

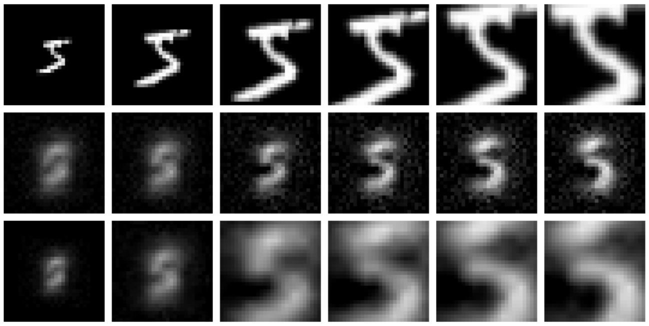

Fig. 5. Results for a training image of “5”: Dilated input training images (first row), images decoded from descriptors (second row) and decoded images dilated with estimated scales (third row)

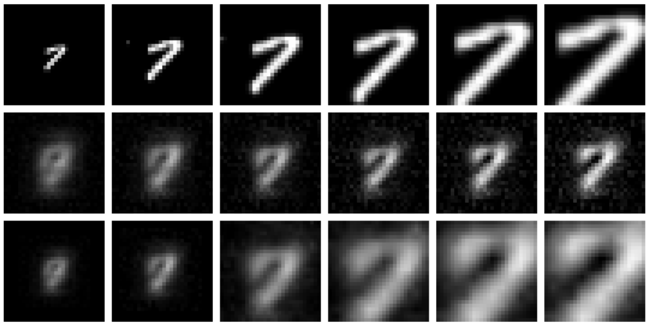

Fig. 6. Results for a training image of “7”: Dilated input training images (first row), images decoded from descriptors (second row) and decoded images dilated with estimated scales (third row)

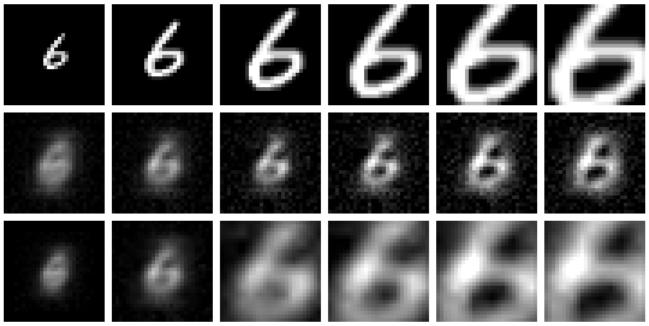

Fig. 7. Results for a training image of “8”: Dilated input training images (first row), images decoded from descriptors (second row) and decoded images dilated with estimated scales (third row)

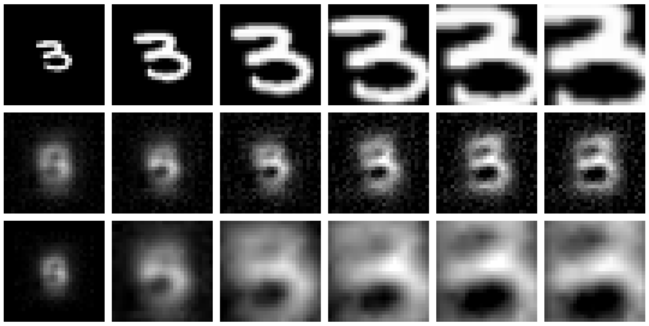

Fig. 8. Results for a test image of "3": Dilated input training images (first row), images decoded from descriptors (second row) and decoded images dilated with estimated scales (third row)

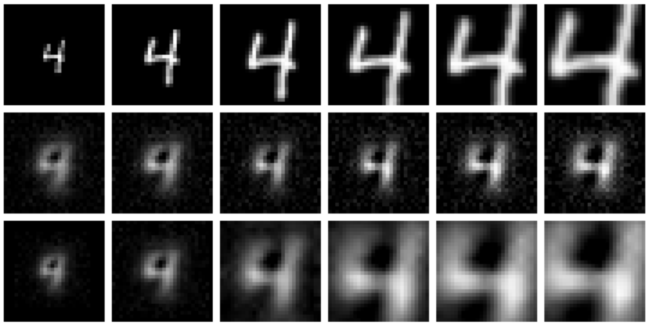

Fig. 9. Results for a test image of "4": Dilated input training images (first row), images decoded from descriptors (second row) and decoded images dilated with estimated scales (third row)

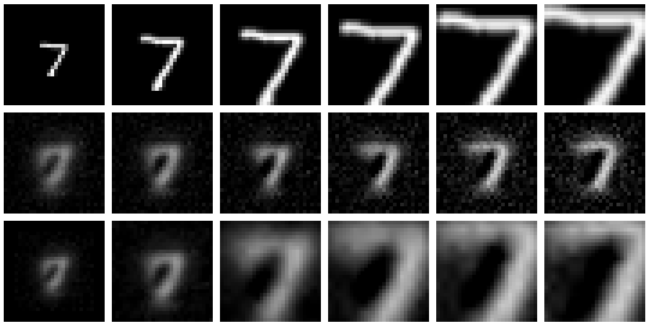

Fig. 10. Results for a test image of "7": Dilated input training images (first row), images decoded from descriptors (second row) and decoded images dilated with estimated scales (third row)

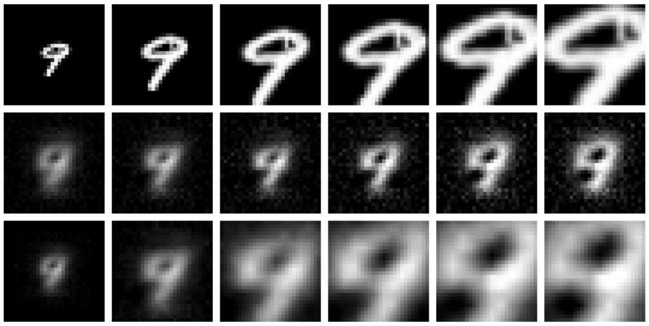

Fig. 11. Results for a test image of "9": Dilated input training images (first row), images decoded from descriptors (second row) and decoded images dilated with estimated scales (third row)

REFERENCES

[1] P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural Networks*, vol. 2, no. 1, pp. 53 – 58, 1989. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0893608089900142

[2] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006. [Online]. Available: http://science.sciencemag.org/content/313/5786/504

[3] A. Makhzani and B. J. Frey, "k-sparse autoencoders," *CoRR*, vol. abs/1312.5663, 2013. [Online]. Available: http://arxiv.org/abs/1312.5663

[4] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 1096–1103. [Online]. Available: http://doi.acm.org/10.1145/1390156.1390294

[5] T. Matsuo, H. Fukuhara, and N. Shimada, "Transform invariant auto-encoder," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017*. IEEE, 2017, pp. 2359–2364. [Online]. Available: https://doi.org/10.1109/IROS.2017.8206047

[6] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.