Transform Invariant Auto-encoder

Tadashi Matsuo, Hiroya Fukuhara,
Nobutaka Shimada

Ritsumeikan University

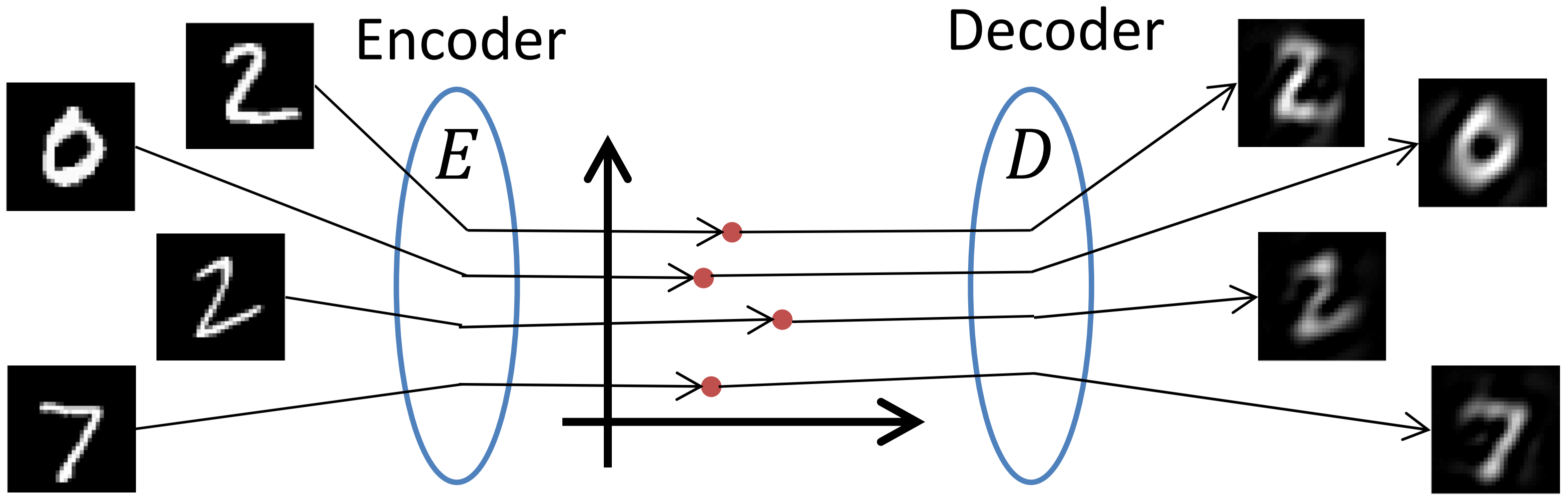
RITSUMEIKAN

Beyond Borders

Auto-encoder

RITSUMEIKAN

A method reducing dimensionality with keeping restoration accuracy



Input image I
32×32

Descriptor $E(I)$
30

Restored image $D\big(E(I)\big)$
32×32

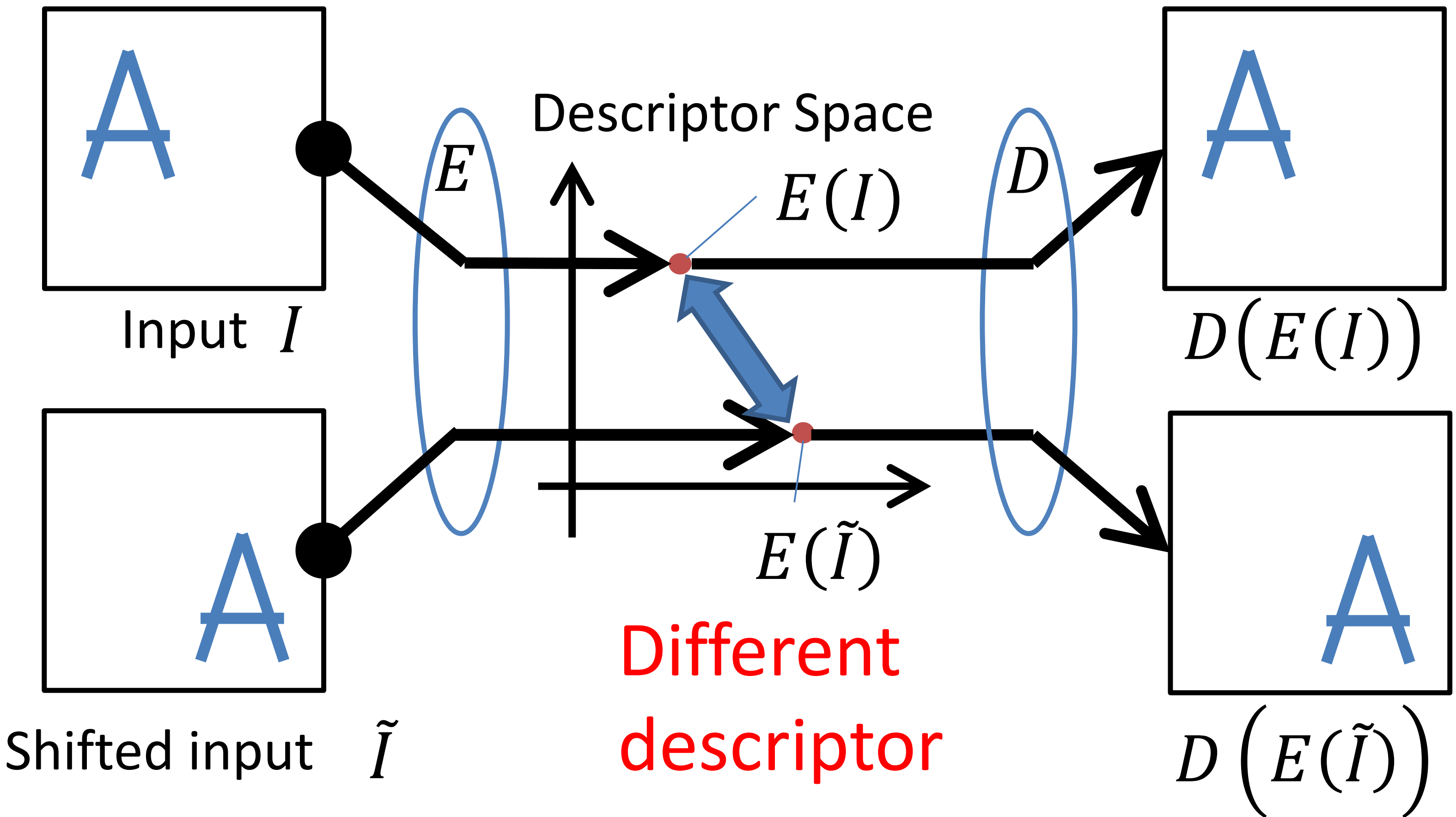
E and D can be trained by minimizing **restoration error** without teacher labels.

$$\sum_I \left\| D\big(E(I)\big) - I \right\|_{L2}^2$$

A descriptor represents essence of an input.

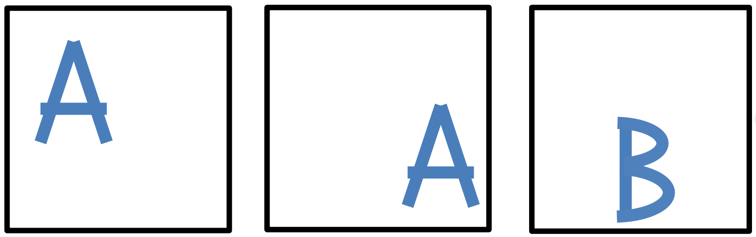
Shift variance of ordinary auto-encoder

Each descriptor implicitly includes shape information.
It is inseparably mixed with positional information.



Same shape

A problem of ordinary auto-encoder



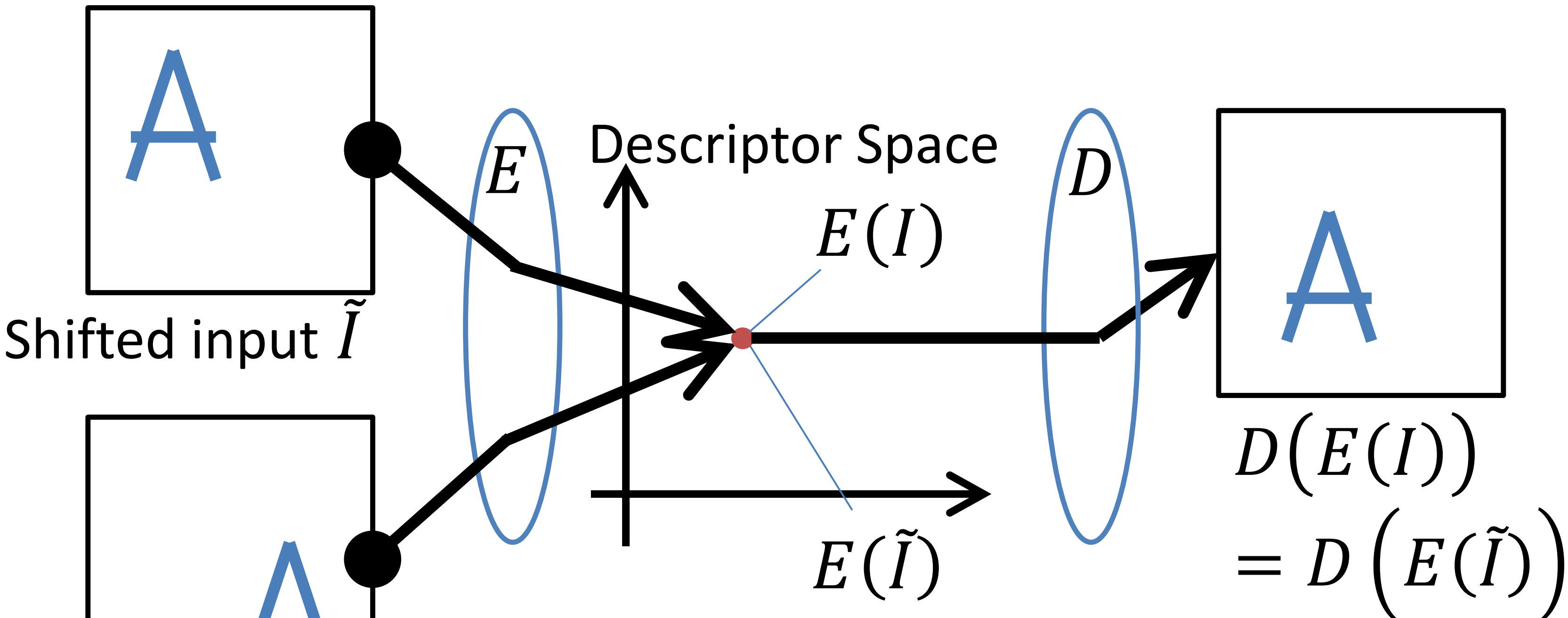
These descriptors differ from each other. To distinguish shapes by descriptors, prior normalization is required.

But, prior normalization may be difficult.

Can we assign similar descriptors to images with common shape without prior normalization?

If an auto-encoder is **shift invariant**, a descriptor represents a shape itself without regard to its position.

Shift invariant auto-encoder

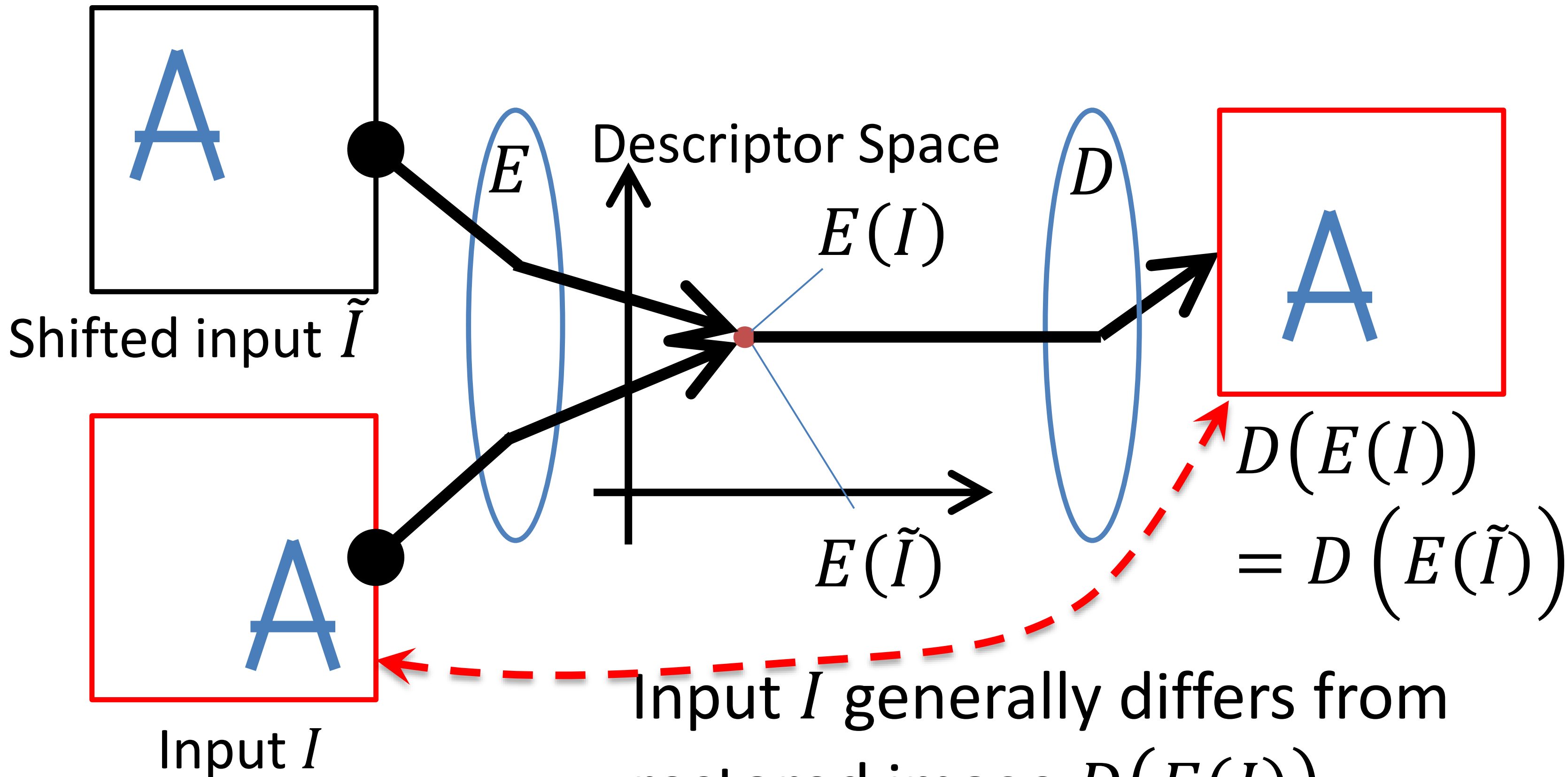


Similar shapes

Similar descriptors

Restored shape

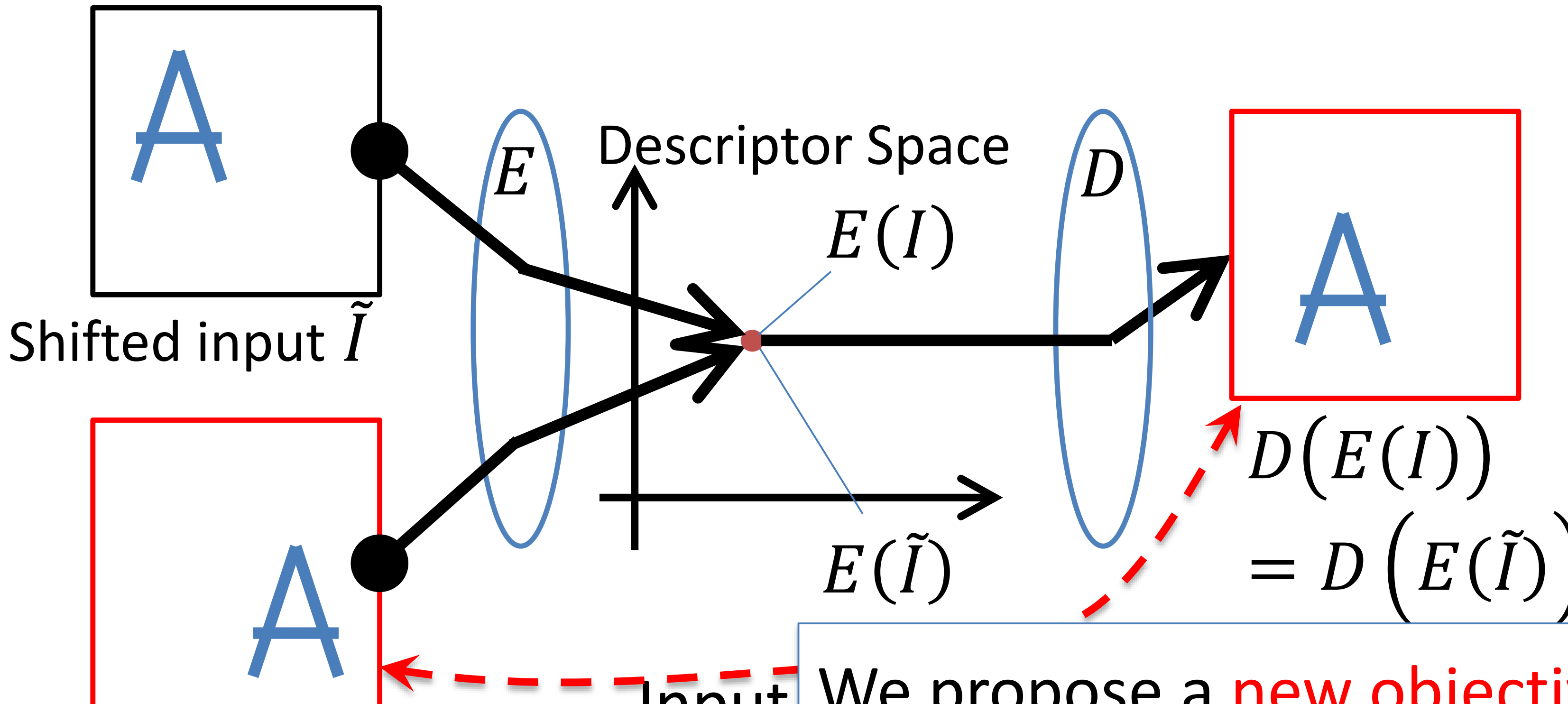
Shift invariant auto-encoder



Input I generally differs from restored image $D(E(I))$.

It cannot be trained with ordinary objective function $\sum_I \|D(E(I)) - I\|_{L2}^2$

Shift invariant auto-encoder



Input

restor

We propose a new objective function that brings

- **shift invariance**,
- **shape restoration**.

It cannot be trained wi

ordinary objective functi

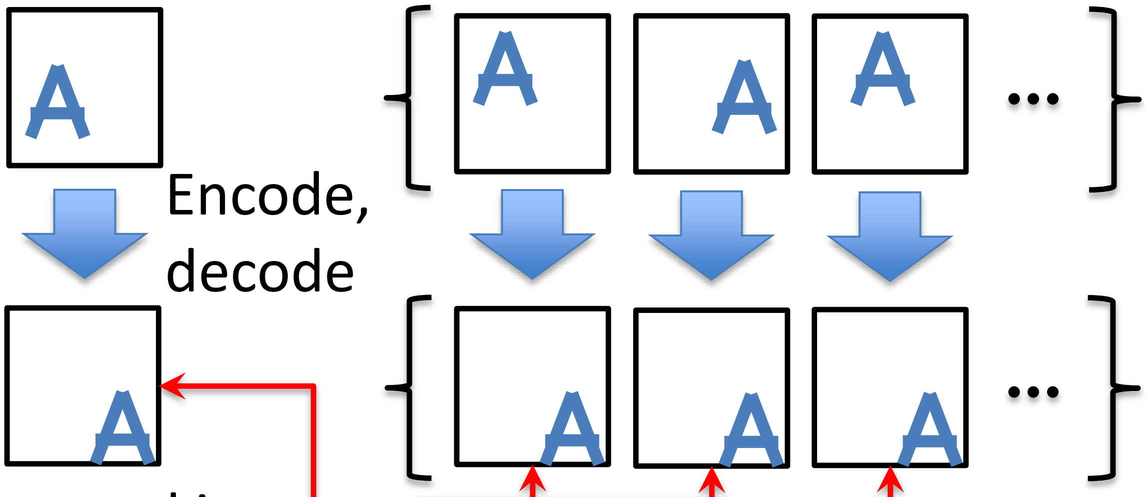
I

Evaluation of shift invariance

Input I

Shifted variations generated from I

T_θ is a shift operator.



$= \{T_\theta(I)\}_\theta$

$= \left\{D\left(E\left(T_\theta(I)\right)\right)\right\}_\theta$

Restored image

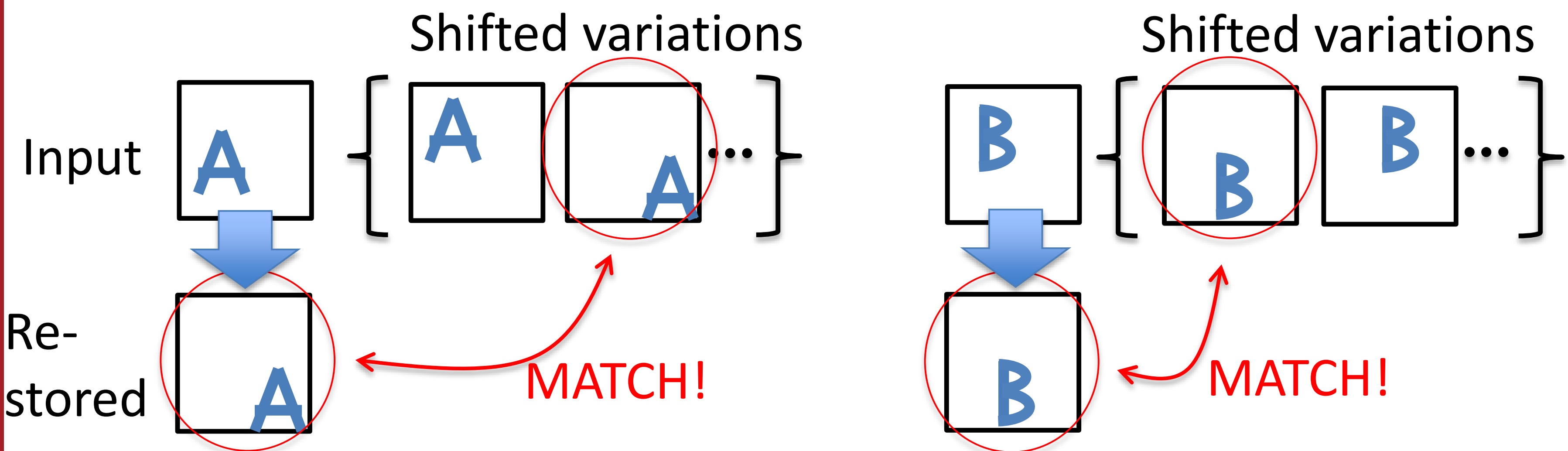
$D\left(E(I)\right)$

All restored images should be similar
if the auto-encoder is shift invariant.

$$\sum_I \sum_\theta \left\| D\left(E(I)\right) - D\left(E\left(T_\theta(I)\right)\right) \right\|_{L2}^2$$

Train E, D so that they minimize this term.

Evaluation of restoration



A restored image should match with one of shifted images

if the auto-encoder can restore a shape.

$$\sum_{I} \left\| D\left(E(I)\right) - T_{\hat{\theta}(I)}(I) \right\|_{L2}^{2}$$

Train E, D so that they minimize this term.

where $\hat{\theta}(I)$ means the "best" transform parameter:

$$\hat{\theta}(I) = \arg\min_{\theta} \left\| D\left(E(I)\right) - T_{\theta}(I) \right\|_{L2}^{2}$$

Proposed objective function

$$C(E,D) = \lambda_{inv} C_{inv}(E,D) + \lambda_{res} C_{res}(E,D) + \lambda_{spa} C_{spa}(E)$$

Invariance term $C_{inv}(E,D)$

$$\sum_I \sum_\theta \left\| D(E(I)) - D\left(E(T_\theta(I))\right) \right\|_{L2}^2$$

Restored image should be unchanged even if inputs are transformed with any parameter.

Restoration term $C_{res}(E,D)$

$$\sum_I \left\| D(E(I)) - T_{\hat{\theta}(I)}(I) \right\|_{L2}^2$$

Restored image should match with one of transformed images.

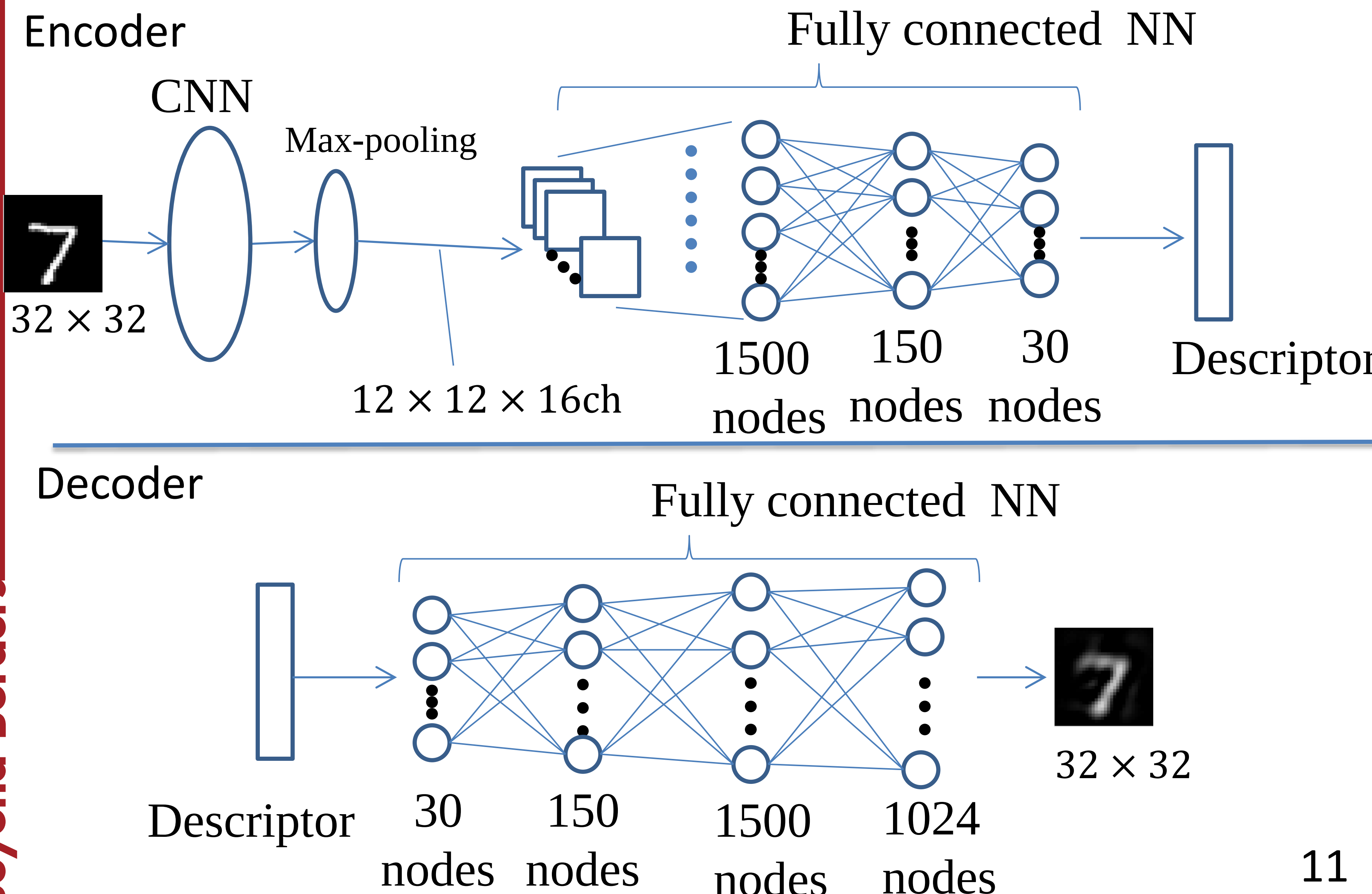
Sparseness term

$$C_{spa}(E)$$

$$\sum_{I \in S} \frac{\|E(I)\|_{L1}^2}{\|E(I)\|_{L2}^2}$$

Descriptor $E(I)$ should be a sparse vector.

Example structure of auto-encoder

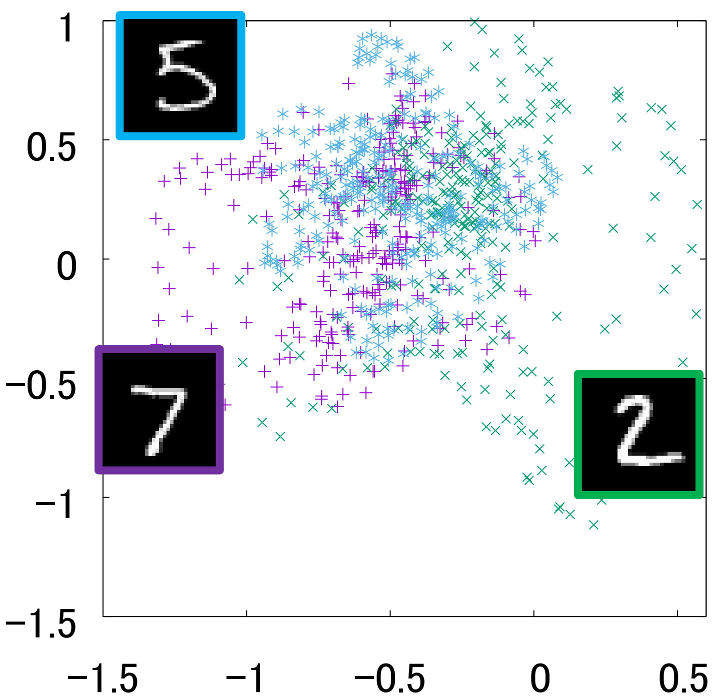


Distribution of descriptors from shift invariant auto-encoder

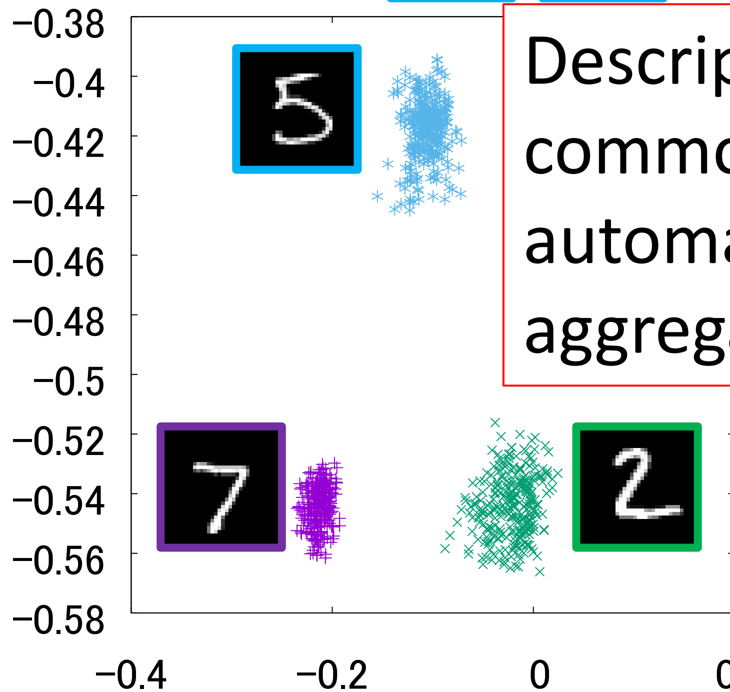
We trained auto-encoders with shifted MNIST training images.

Input:32 × 32
Descriptor dim: 30
Max shift width: 8

Distributions of descriptors from shifted test images such as .



Ordinary auto-encoder



Descriptors from common shapes automatically aggregate.

Shift invariant auto-encoder

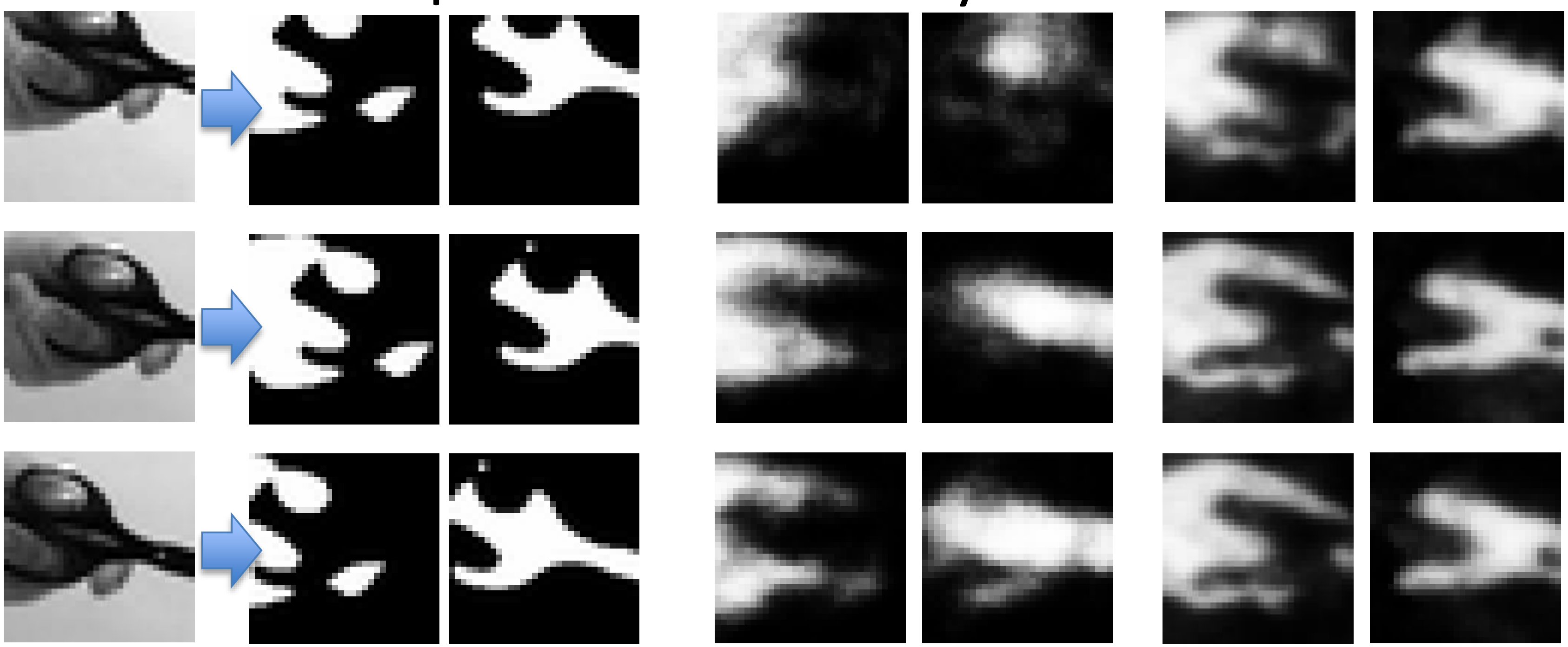
Example for hand-object interaction

Situation | Hand mask | Object mask

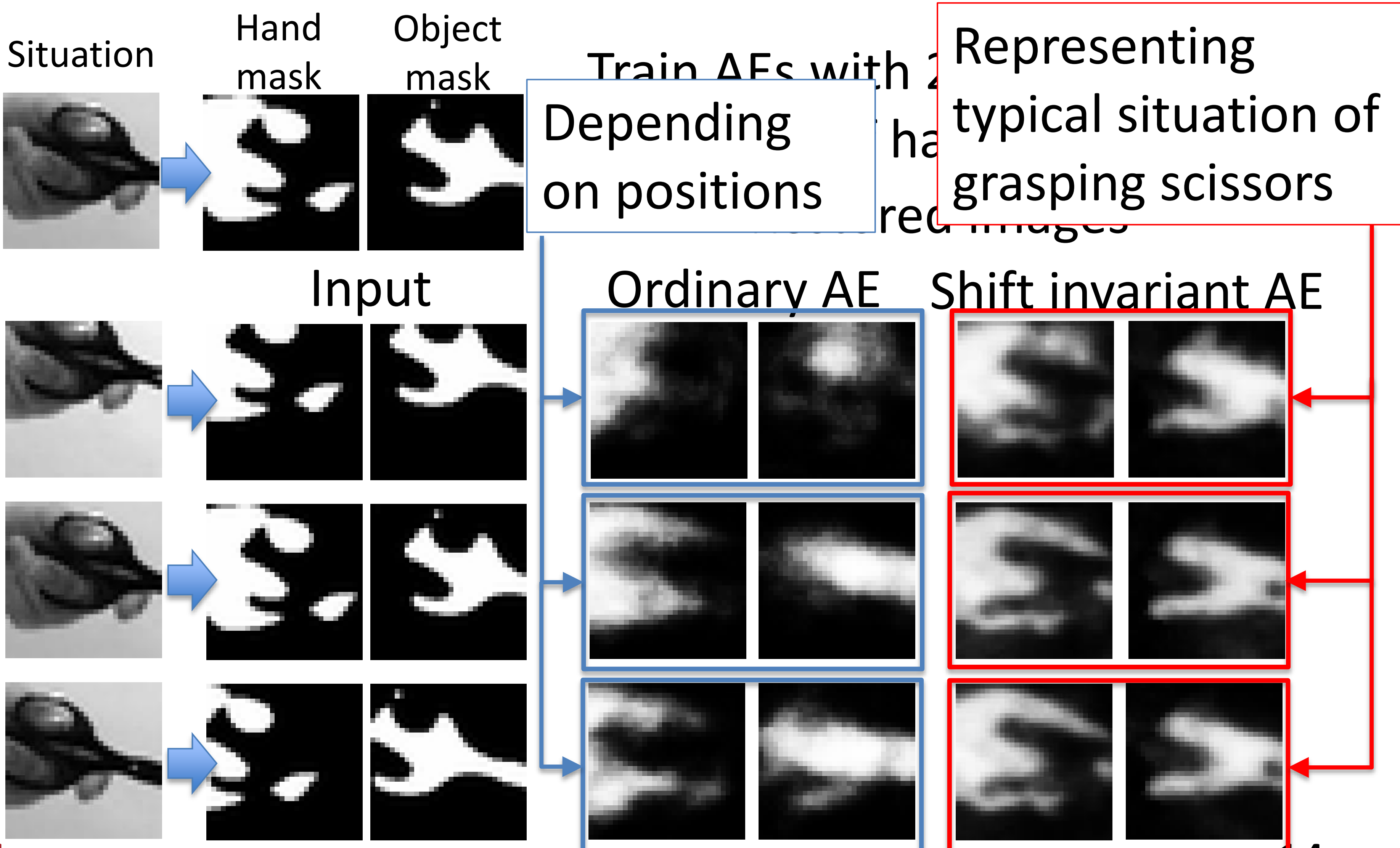
Train AEs with 2-channel images consisting of hand/object masks.

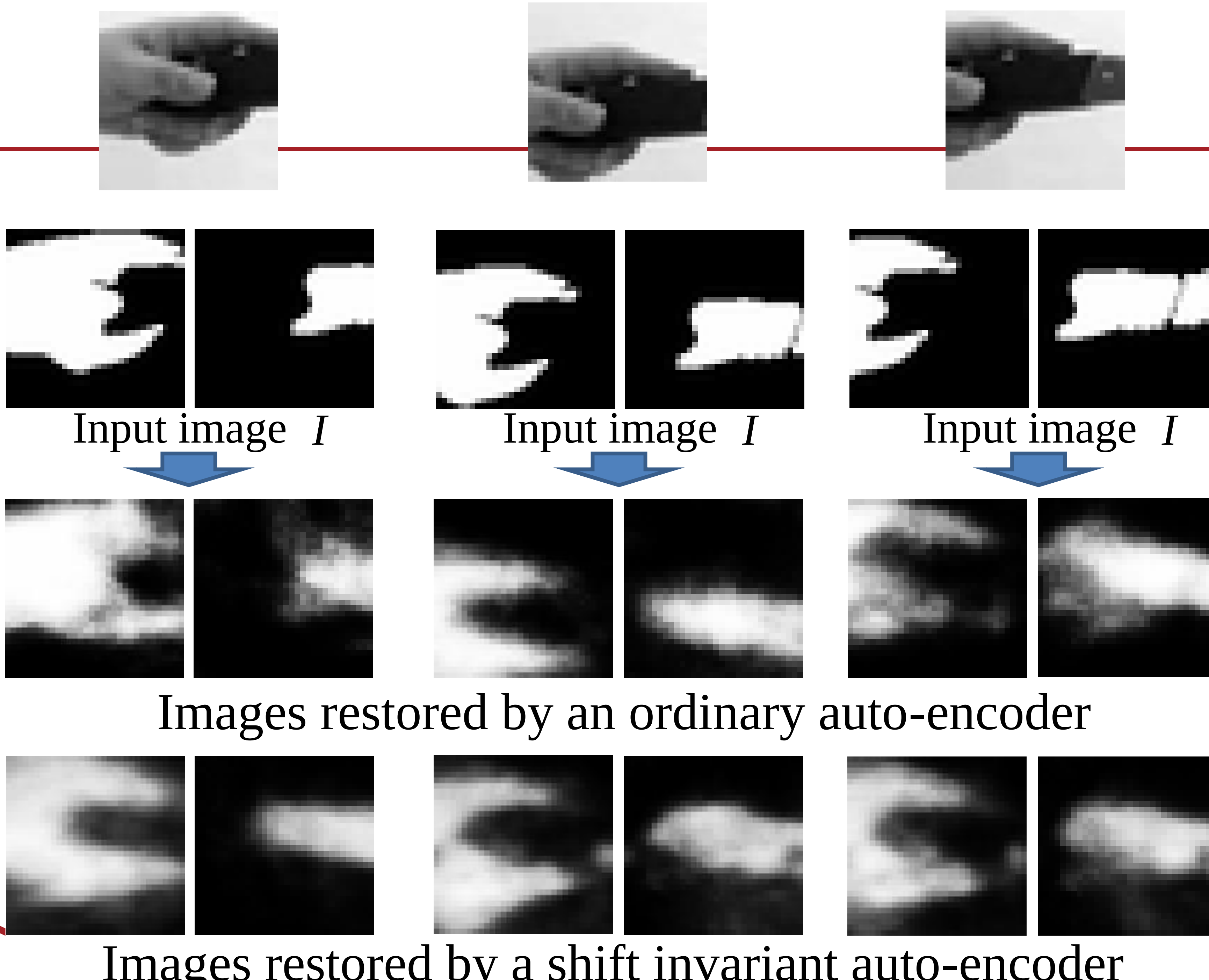
Restored images

Input | Ordinary AE | Shift invariant AE

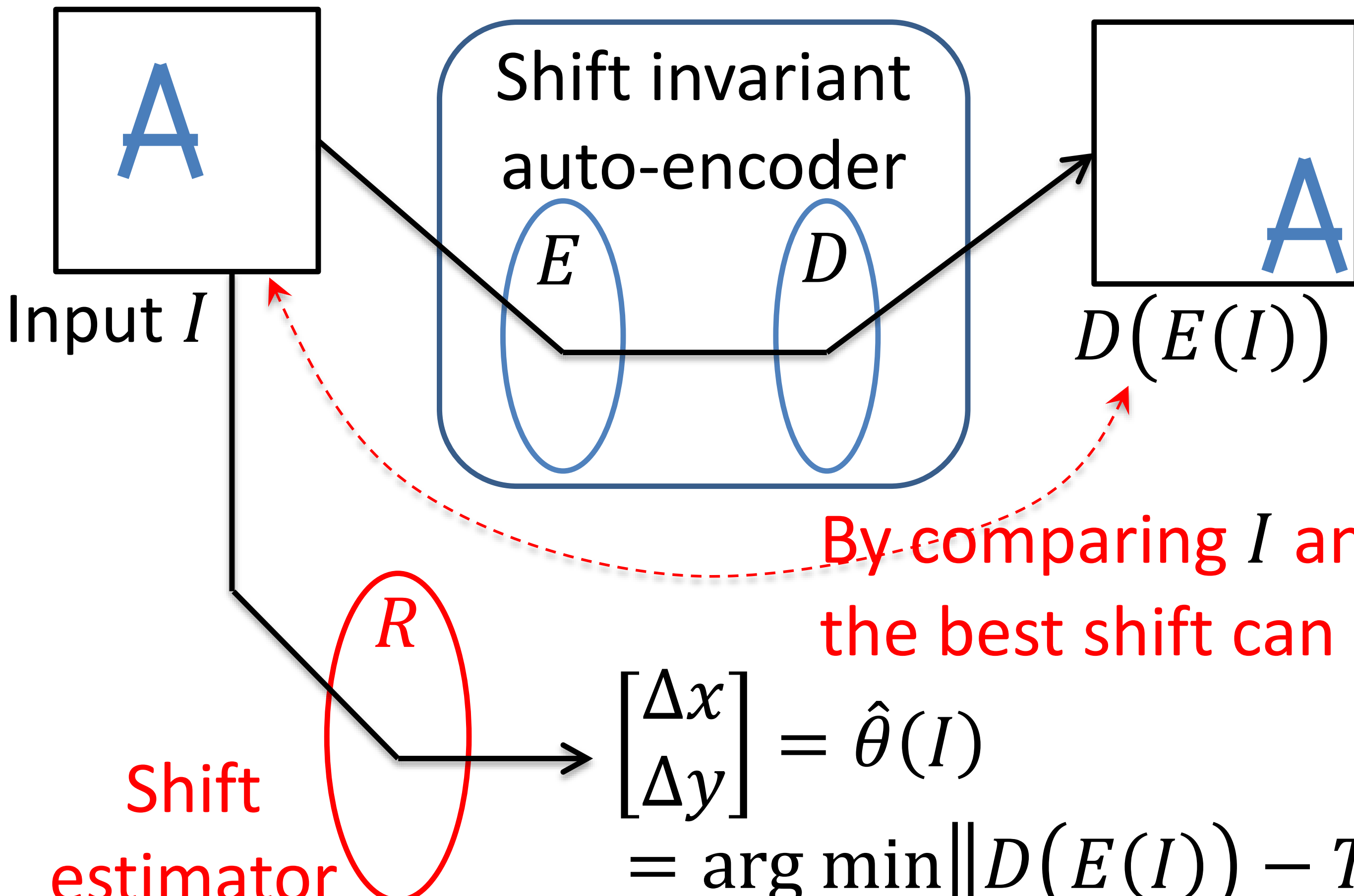


Example for hand-object interaction




Input image I
Input image I
Input image I
Images restored by an ordinary auto-encoder
Images restored by a shift invariant auto-encoder

Estimation of shift parameter



By comparing I and $D(E(I))$, the best shift can be calculated.

$$\begin{bmatrix}\Delta x \\ \Delta y\end{bmatrix} = \hat{\theta}(I)$$

$$= \arg\min_{\theta} \left\| D(E(I)) - T_{\theta}(I) \right\|_{L2}^{2}$$

We can obtain shift estimator by training neural network with pairs of an input and its best shift.

Example of decomposition by shift invariant auto-encoder

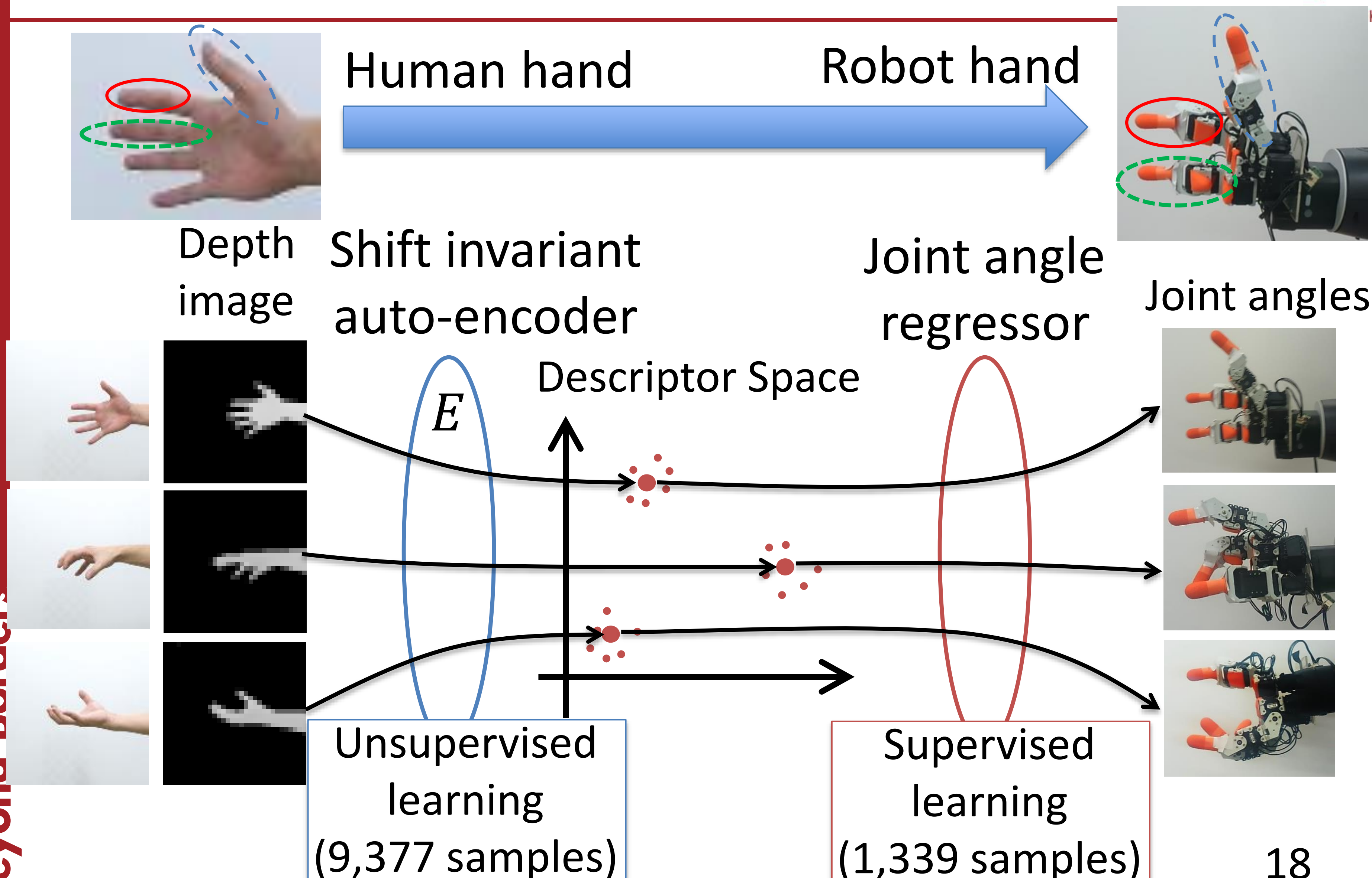
Inputs | Restored images | Restored images shifted by estimated shifts

$D(E(I))$ — Shape only

$T_{R(I)}\left(D(E(I))\right)$ — Position and shape

An input I can be decomposed into shape descriptor $E(I)$ and shift parameter $R(I)$.

Example of human hand imitation



Beyond Borders

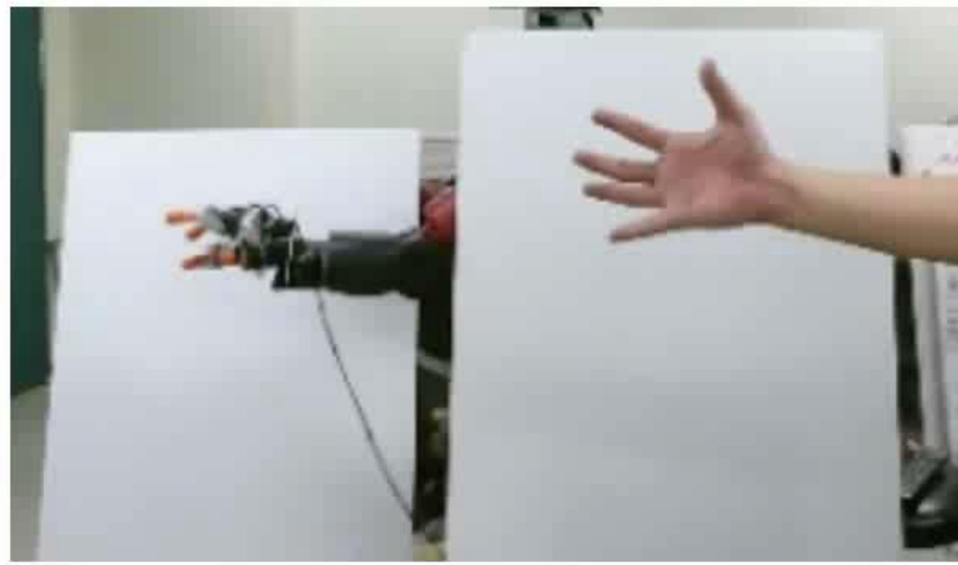
Example of human hand imitation

Input
depth image

Restored
depth image

Imitating
robot hand

Input
human hand

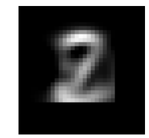


Even if the hand moves, restored images are fixed because of shift invariance.

Joint angles of imitating robot hand were successfully estimated from the descriptors.

Conclusion

We proposed transform invariant auto-encoder that can extract invariants as a descriptor.

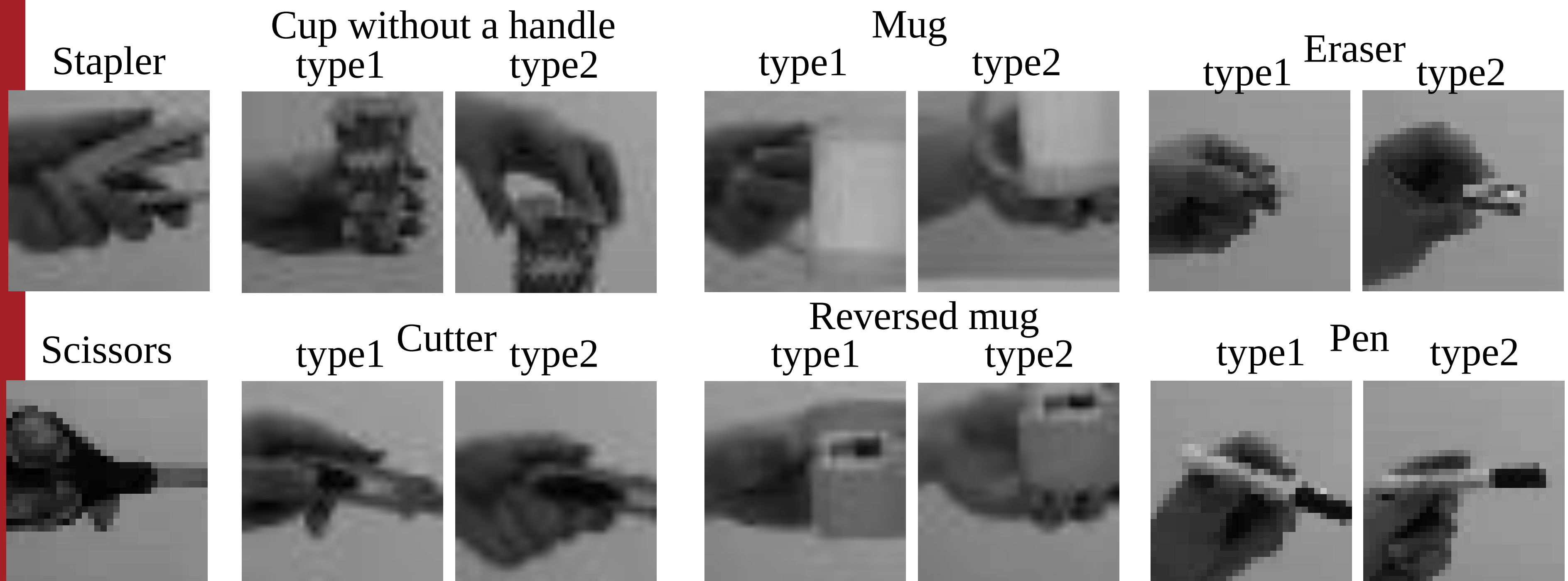
Shift  Shape is invariant. 

- The invariance is realized by objective function on training.
- An input can be decomposed into invariant components and variant components.

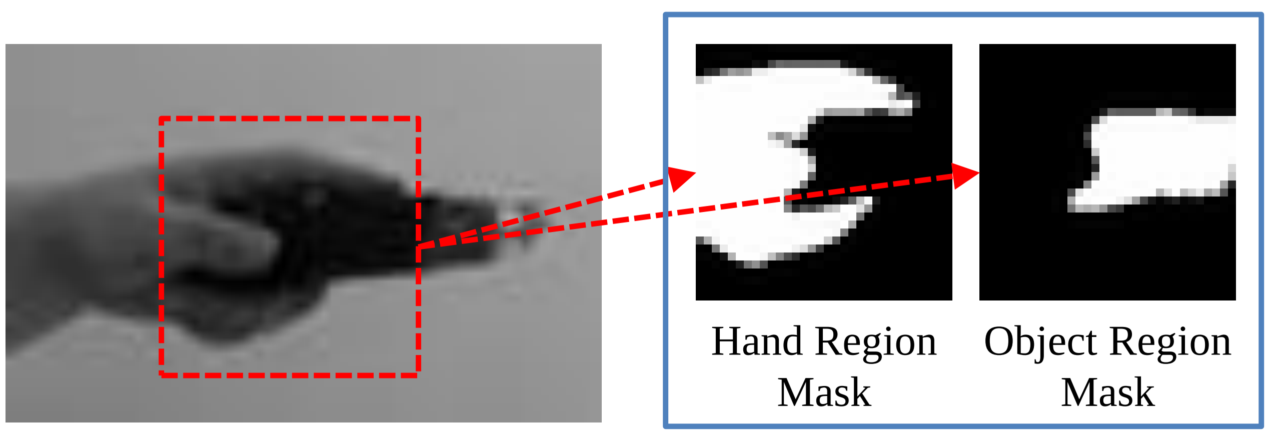
Future work

- Computation cost for multiple types of transform
- Applying other transforms, such as scaling, rotation and blurring

Trained interactions

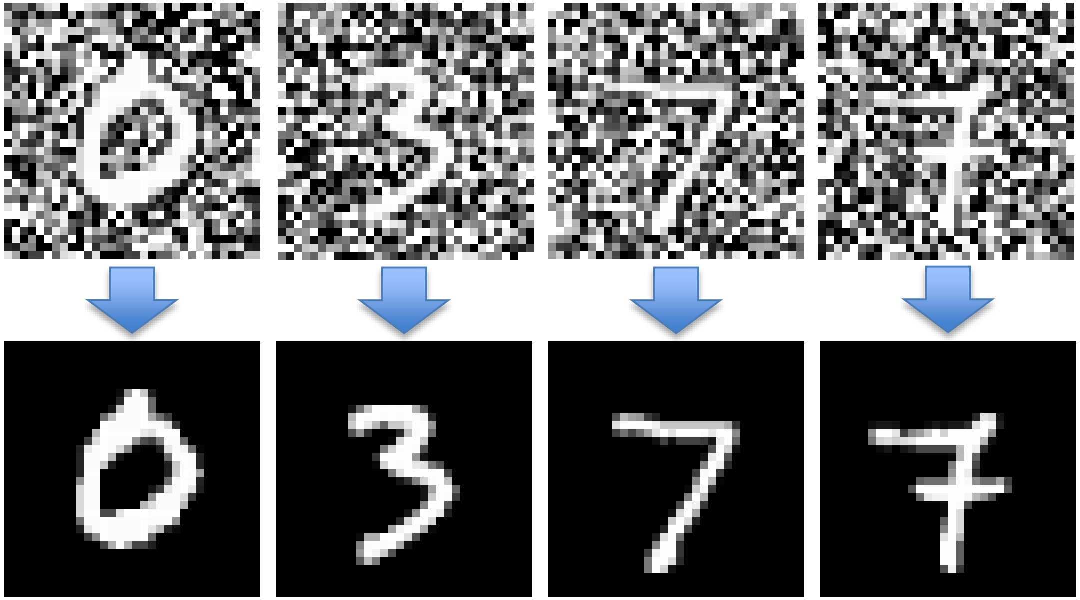


1680 scenes for training



Interaction Image (32*32*2ch)

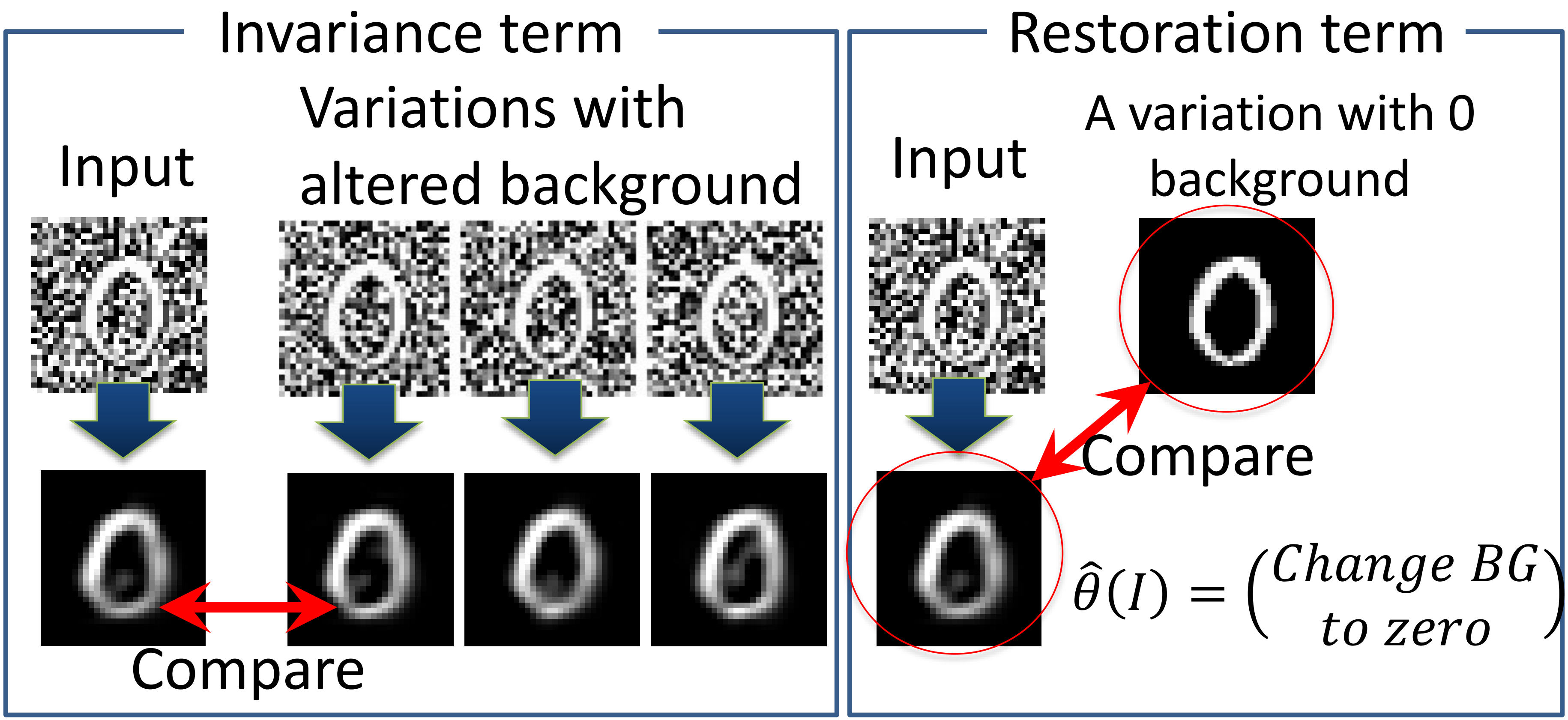
Extraction of foreground pattern from complex background



The proposed method can be applied to extraction of foreground pattern.
Foreground patterns have many variation like hand-written numbers.

Foreground auto-encoder

By considering invariance to background alteration, we can generate a descriptor of foreground pattern itself.



Objective function for foreground auto-encoder

$$C(E,D) = \lambda_{inv} C_{inv}(E,D) + \lambda_{res} C_{res}(E,D) + \lambda_{spa} C_{spa}(E)$$

Invariance term $C_{inv}(E,D)$

$$\sum_{I} \sum_{\theta} \left\| D(E(I)) - D\left(E(T_{\theta}(I))\right) \right\|_{L2}^{2}$$

Restored image should be unchanged even if inputs are transformed with any parameter.

Input I

$\{T_{\theta}(I)\}$

$=$

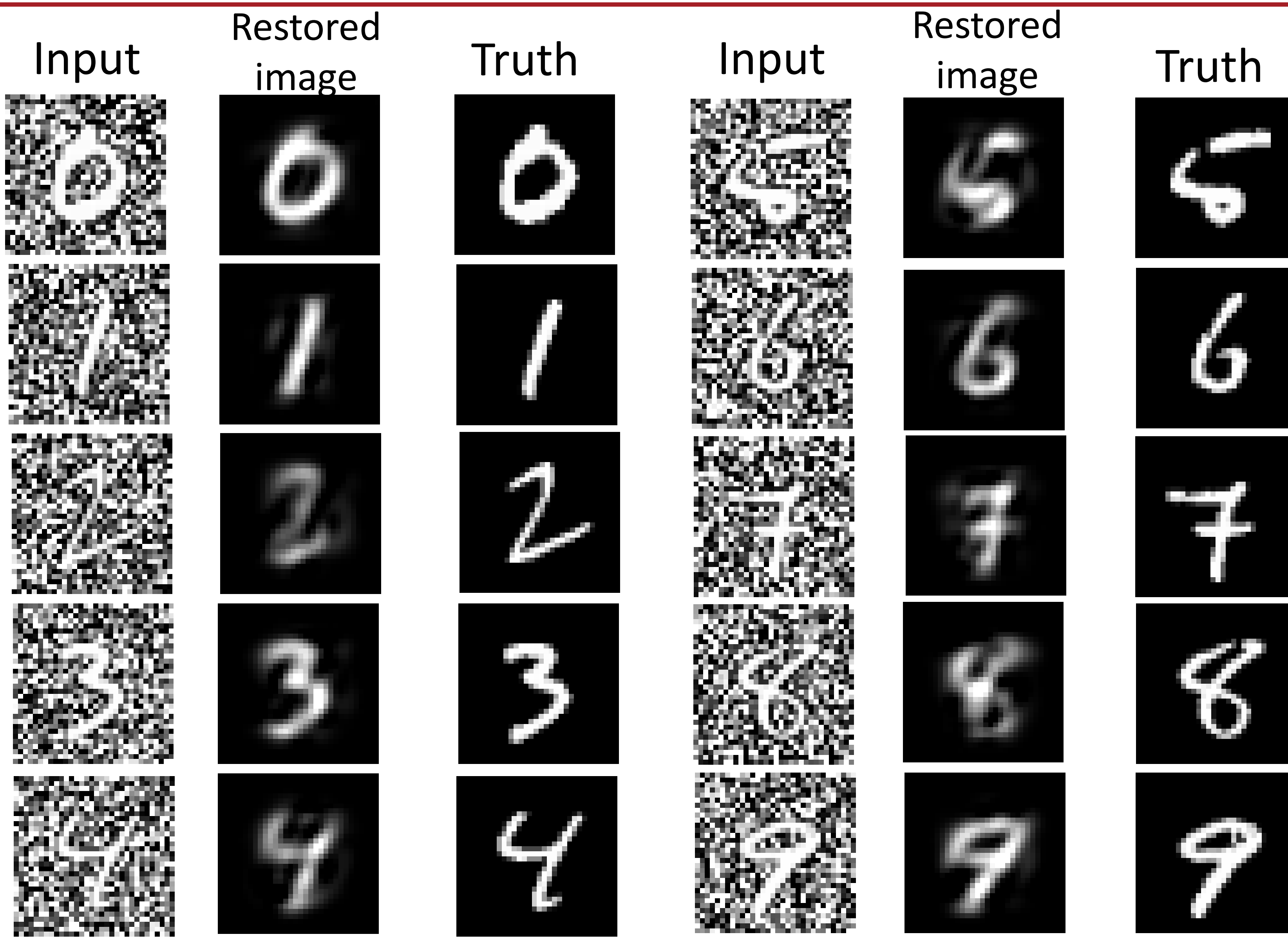
Restoration term $C_{res}(E,D)$

$$\sum_{I} \left\| D(E(I)) - T_{\hat{\theta}(I)}(I) \right\|_{L2}^{2}$$

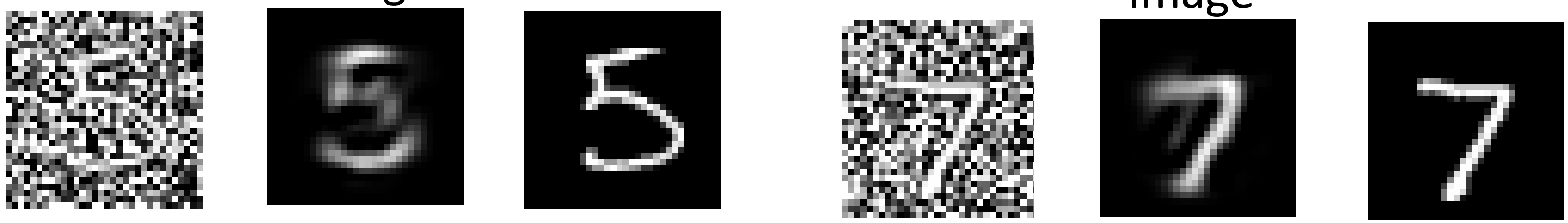
Restored image should match with the variation with 0-background.

$T_{\hat{\theta}(I)}(I) =$ 

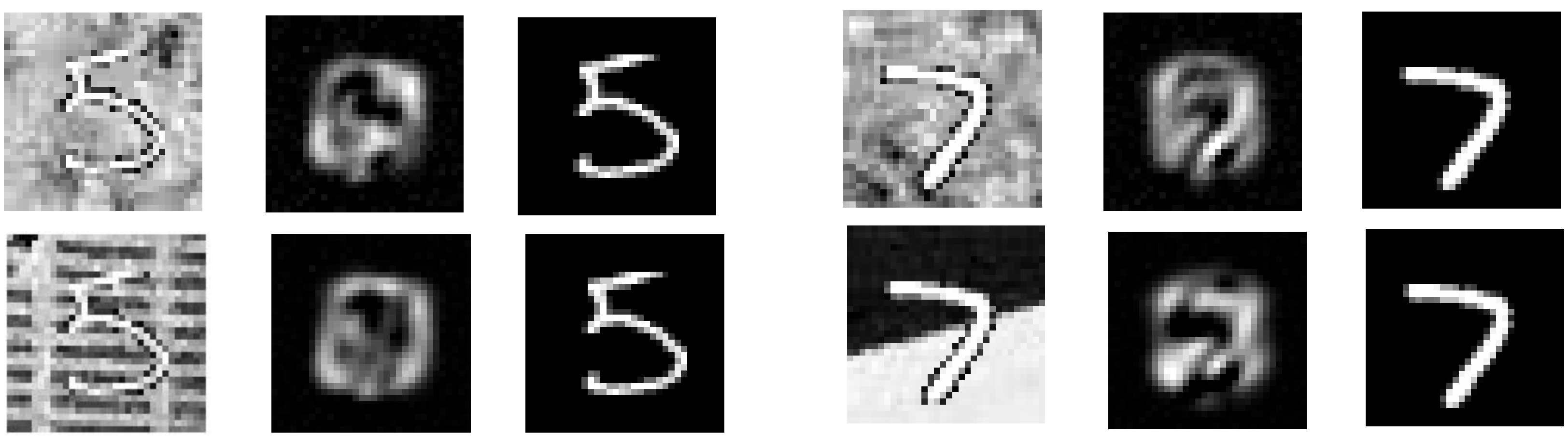
Example of foreground extraction



Dependency of background

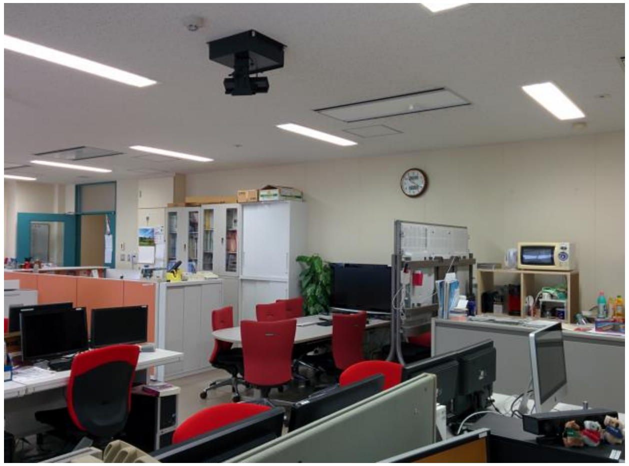


Results for images with indoor scene background



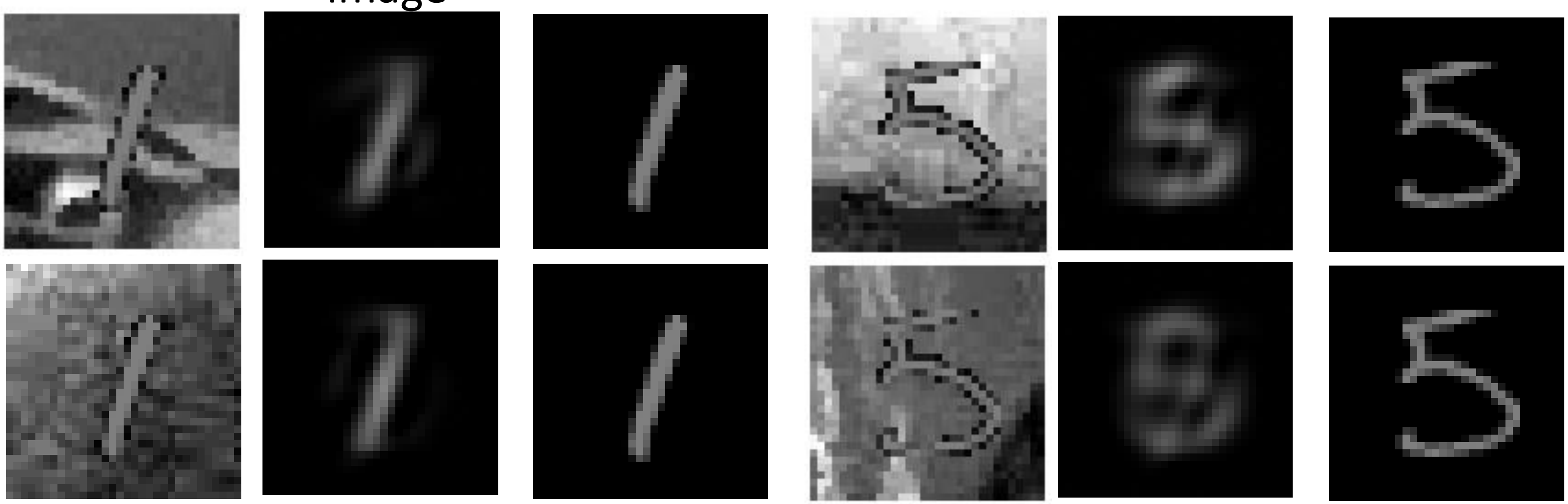
An encoder trained with noisy background cannot extract images with indoor scene background.

Example for indoor background



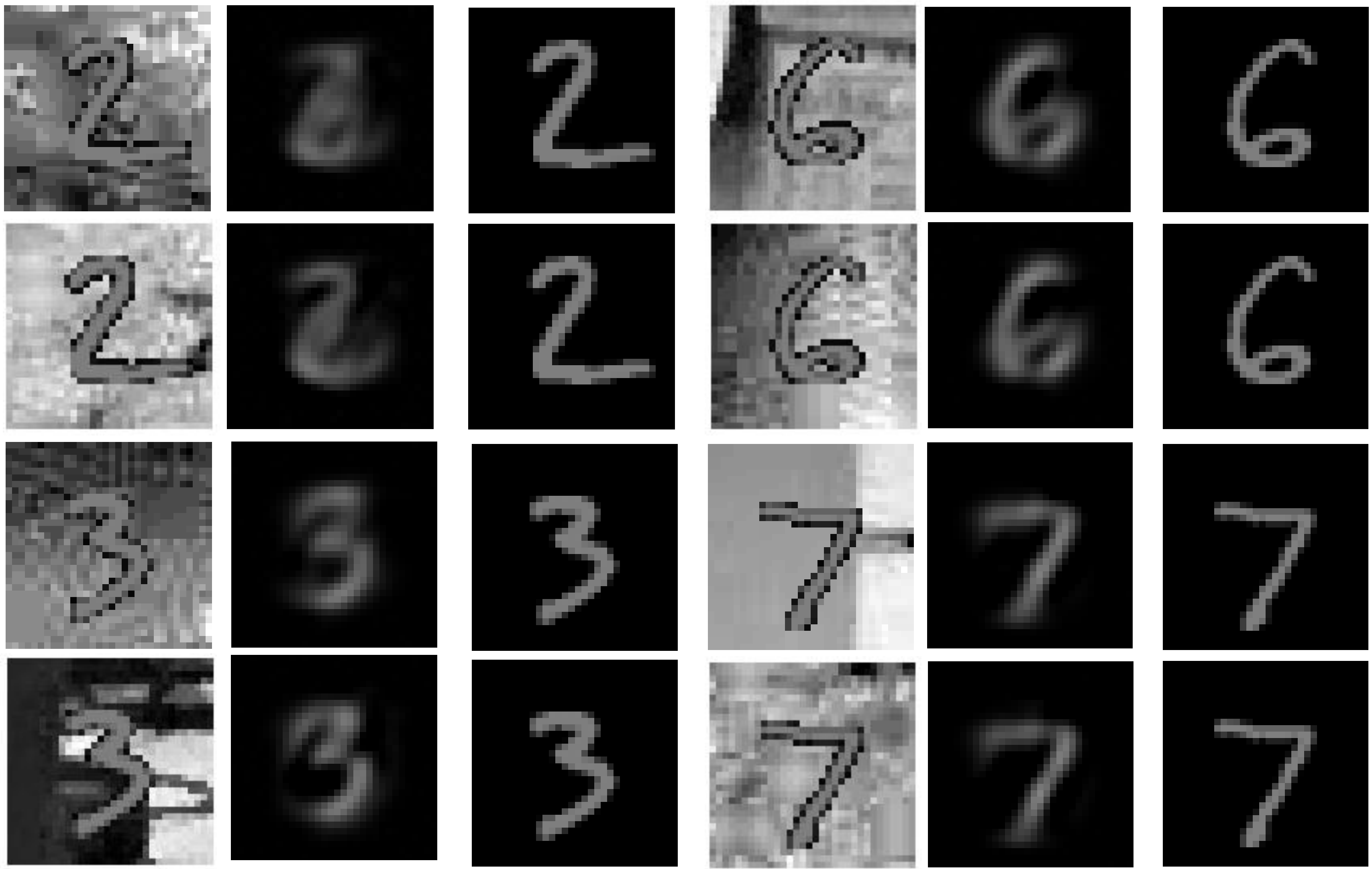
Use 32×32 images extracted from an indoor scene image (3200×2368) as background variation.

Input	Restored image	Truth	Input	Restored image	Truth



Extraction from images with indoor scene background

Input | Restored image | Truth | Input | Restored image | Truth



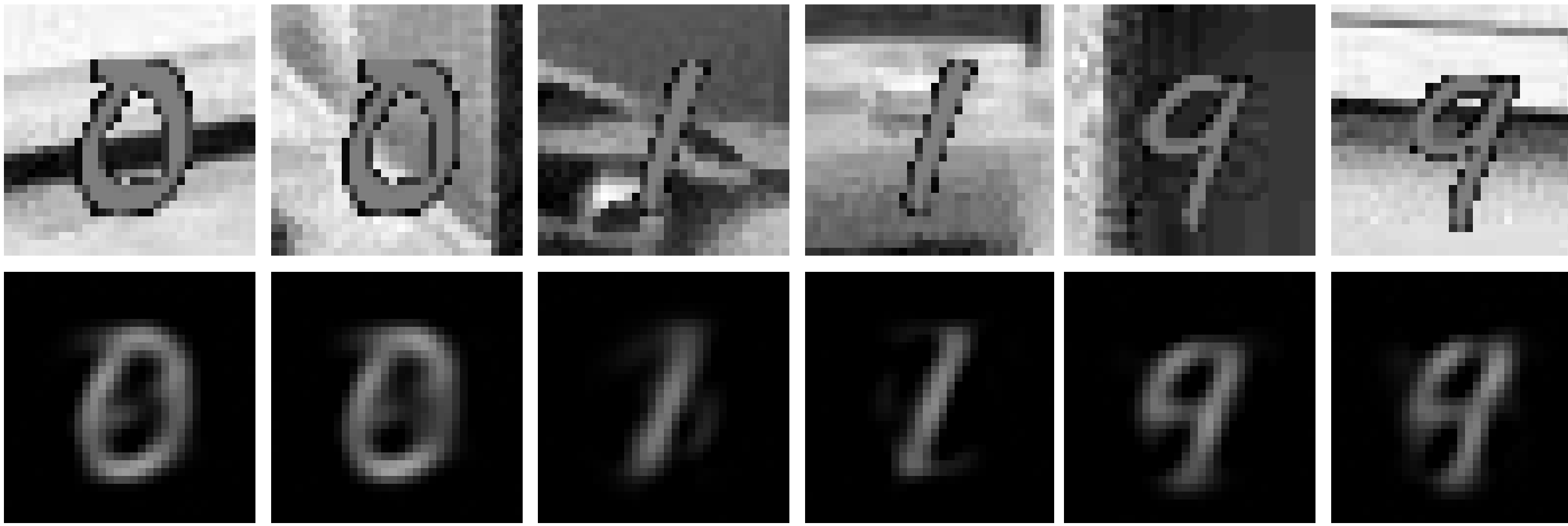
Beyond Borders

RITSUMEIKAN

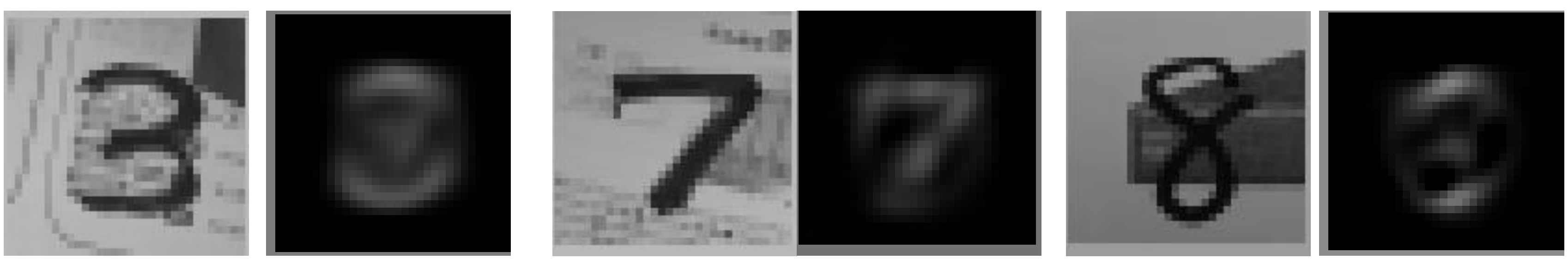
Extraction from images with indoor scene background

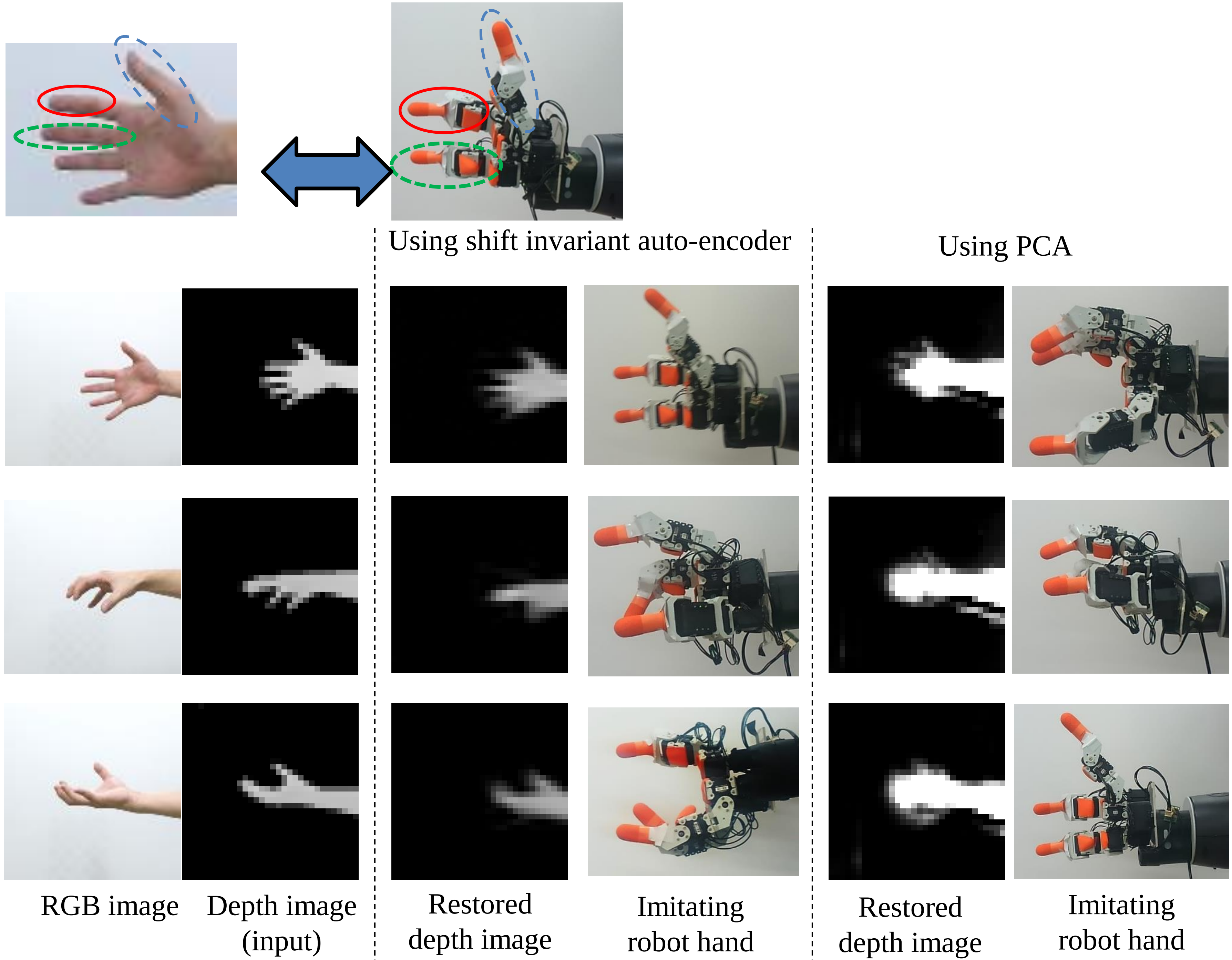
Input

Restored image

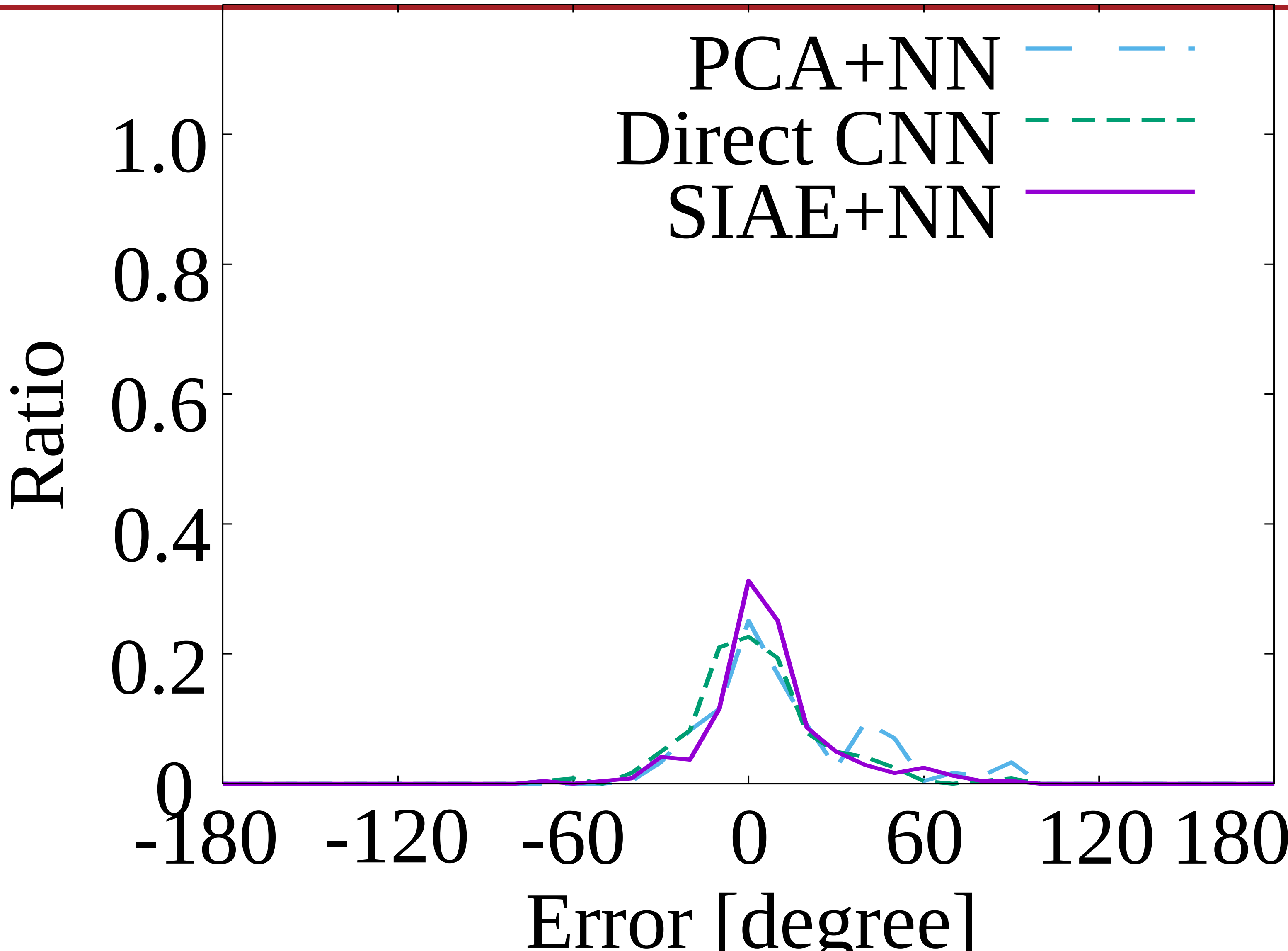


Hand written number on flyer

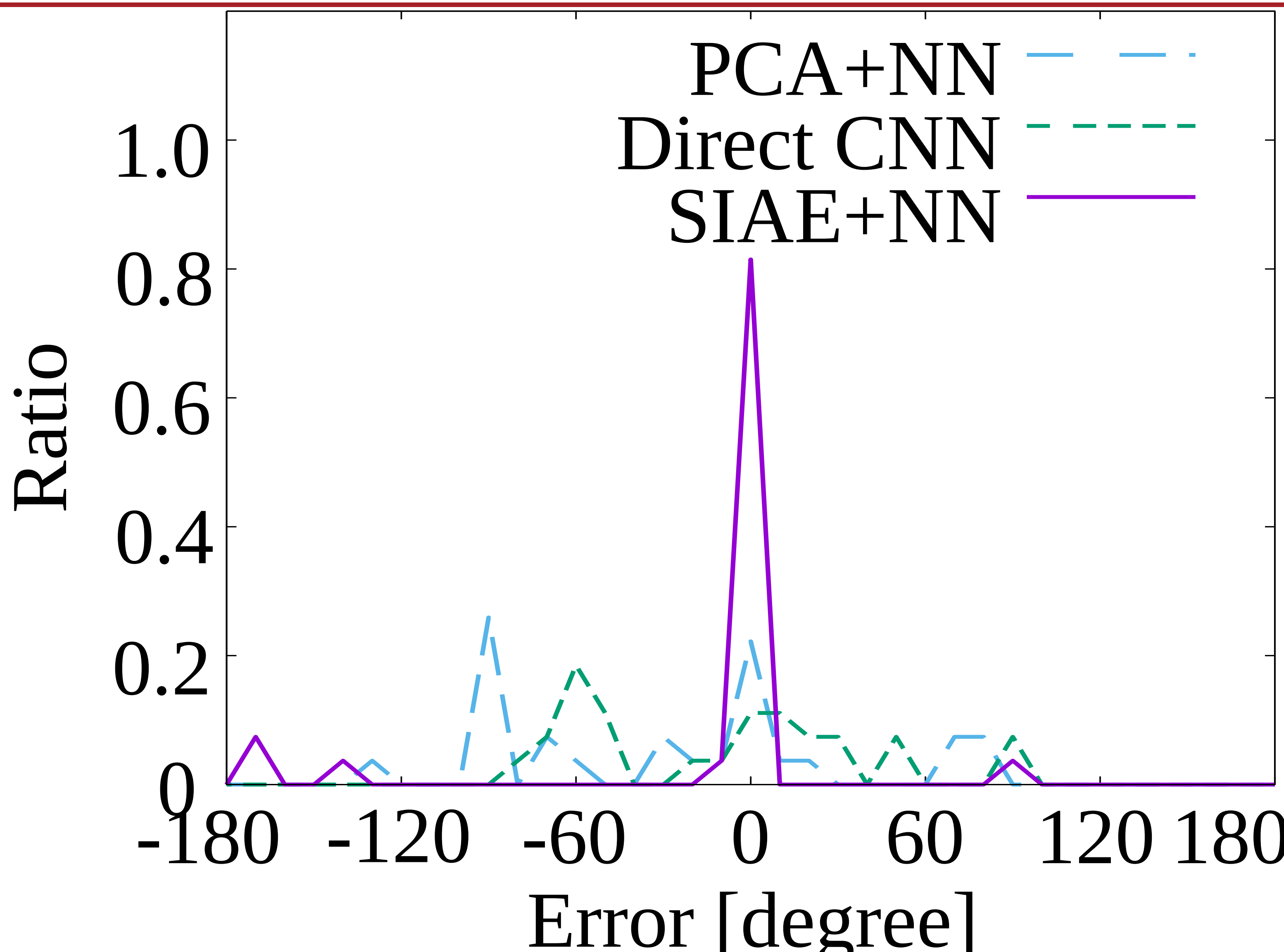



Using shift invariant auto-encoder
Using PCA
RGB image
Depth image (input)
Restored depth image
Imitating robot hand
Restored depth image
Imitating robot hand

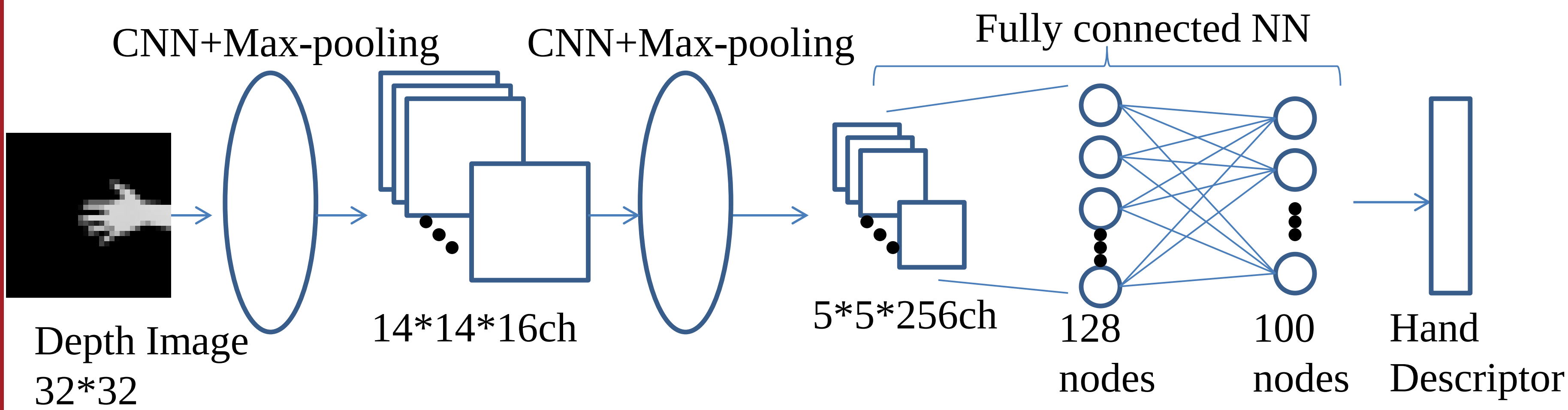
Error of angles of fingers

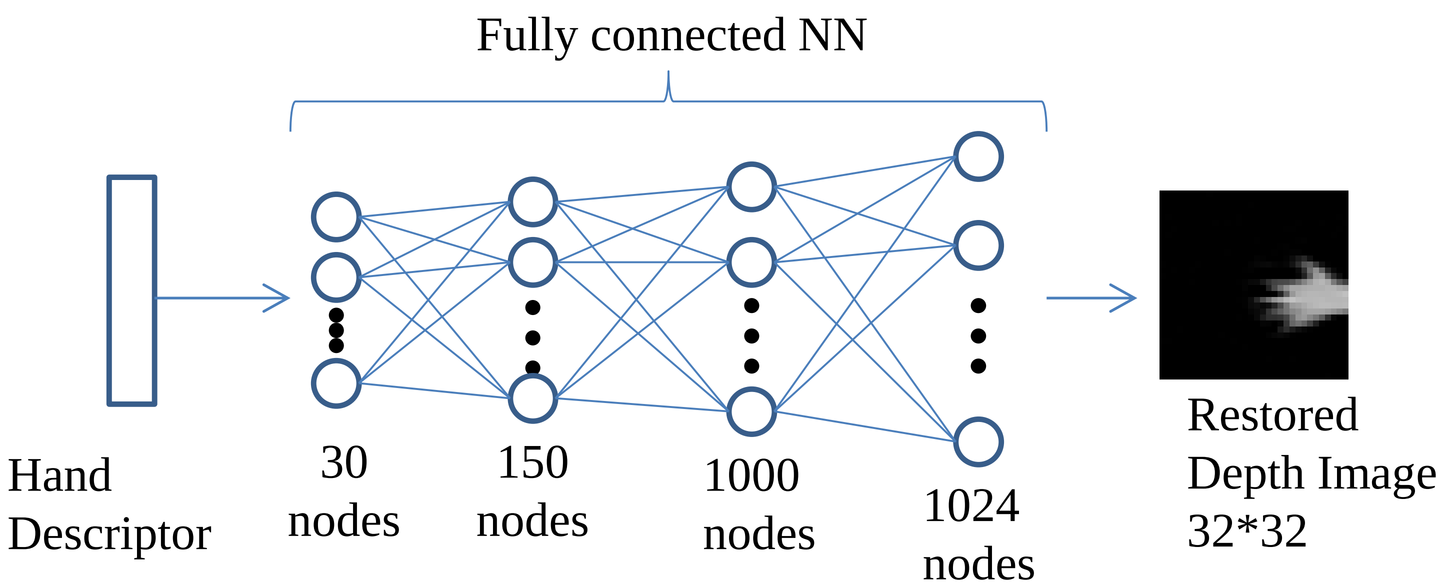


Error of wrist angles



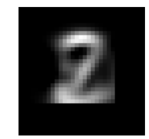
Structure of auto-encoder for hand descriptor

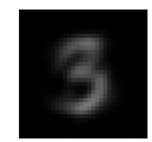




まとめ

変換に関して不変な成分のみを表すdescriptorを出力するauto-encoderを提案

平行移動変換（シフト）  形状は不変 

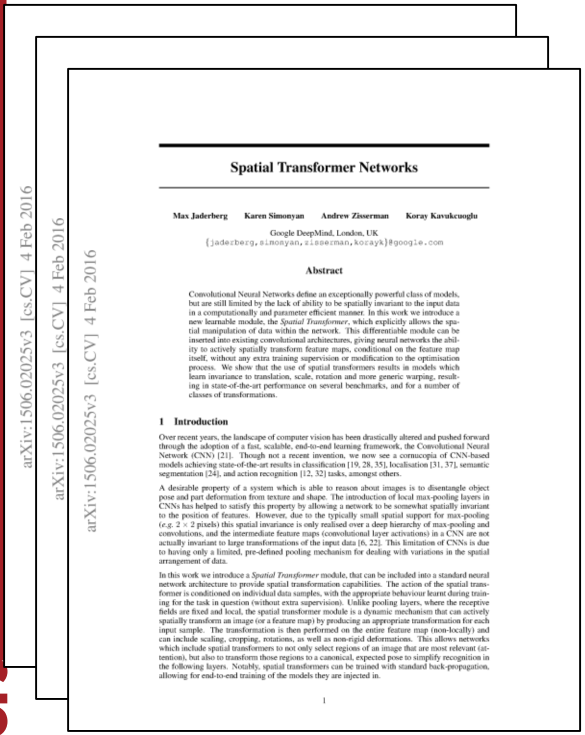
背景変更  前景は不変 

- 目的関数で実現する（特殊なNNは不要）
- 入力を（不変量、変換パラメータ）のペアに分解可能

課題

- 複数種類の変換を組み合わせた際の計算量増加に対する対処
- 拡大縮小や回転、blurなどの変換への適用

Example of document



arXiv:1506.02025v3 [cs.CV] 4 Feb 2016

Spatial Transformer Networks

Max Jaderberg Karen Simonyan Andrew Zisserman Koray Kavukcuoglu

Google DeepMind, London, UK
{jaderberg,simonyan,zisserman,korayk}@google.com

Abstract

Convolutional Neural Networks define an exceptionally powerful class of models, but are still limited by the lack of ability to be spatially invariant to the input data in a computationally and parameter efficient manner. In this work we introduce a new learnable module, the *Spatial Transformer*, which explicitly allows the spatial manipulation of data within the network. This differentiable module can be inserted into existing convolutional architectures, giving neural networks the ability to actively spatially transform feature maps, conditional on the feature map itself, without any extra training supervision or modification to the optimisation process. We show that the use of spatial transformers results in models which learn invariance to translation, scale, rotation and more generic warping, resulting in state-of-the-art performance on several benchmarks, and for a number of classes of transformations.

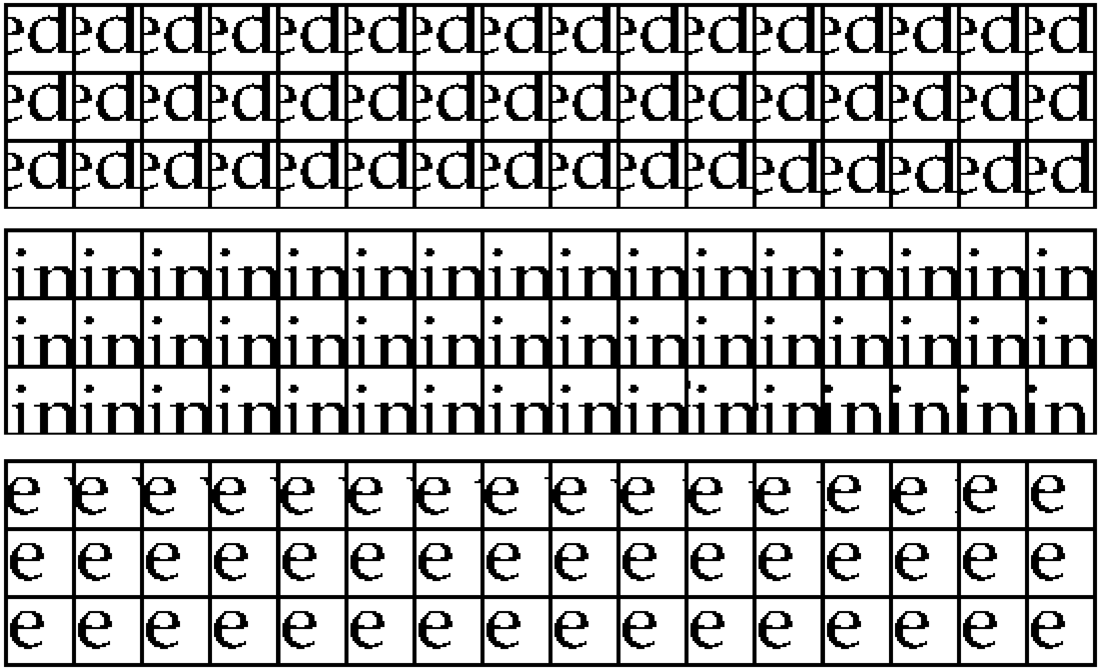
1 Introduction

Over recent years, the landscape of computer vision has been drastically altered and pushed forward through the adoption of a fast, scalable, end-to-end learning framework, the Convolutional Neural Network (CNN) [21]. Though not a recent invention, we now see a cornucopia of CNN-based models achieving state-of-the-art results in classification [19, 28, 35], localisation [31, 37], semantic segmentation [24], and action recognition [12, 32] tasks, amongst others.

A desirable property of a system which is able to reason about images is to disentangle object pose and part deformation from texture and shape. The introduction of local max-pooling layers in CNNs has helped to satisfy this property by allowing a network to be somewhat spatially invariant to the position of features. However, due to the typically small spatial support for max-pooling (*e.g.* 2×2 pixels) this spatial invariance is only realised over a deep hierarchy of max-pooling and convolutions, and the intermediate feature maps (convolutional layer activations) in a CNN are not actually invariant to large transformations of the input data [6, 22]. This limitation of CNNs is due to having only a limited, pre-defined pooling mechanism for dealing with variations in the spatial arrangement of data.

In this work we introduce a *Spatial Transformer* module, that can be included into a standard neural network architecture to provide spatial transformation capabilities. The action of the spatial transformer is conditioned on individual data samples, with the appropriate behaviour learnt during training for the task in question (without extra supervision). Unlike pooling layers, where the receptive fields are fixed and local, the spatial transformer module is a dynamic mechanism that can actively spatially transform an image (or a feature map) by producing an appropriate transformation for each input sample. The transformation is then performed on the entire feature map (non-locally) and can include scaling, cropping, rotations, as well as non-rigid deformations. This allows networks which include spatial transformers to not only select regions of an image that are most relevant (attention), but also to transform those regions to a canonical, expected pose to simplify recognition in the following layers. Notably, spatial transformers can be trained with standard back-propagation, allowing for end-to-end training of the models they are injected in.

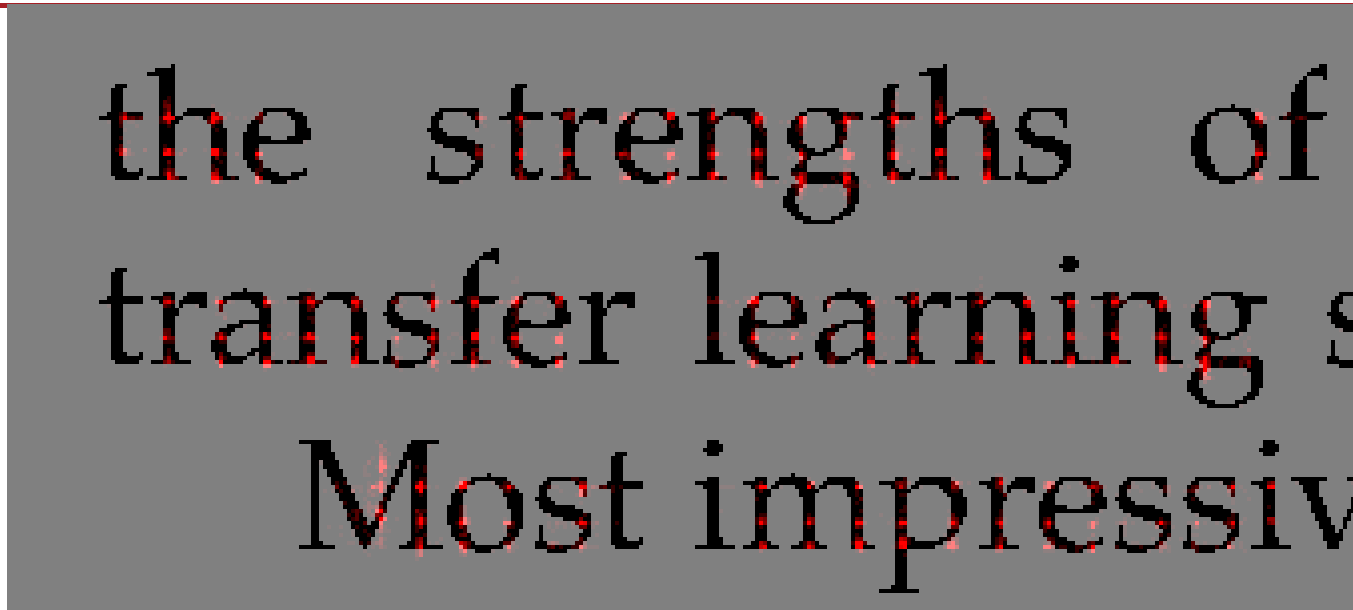
1



By applying mean shift clustering to descriptors, we obtained clusters corresponding to pairs of characters.

[1] Max Jaderberg, Karen Simonyan, Andrew Zisserman and Koray Kavukcuoglu. “Spatial Transformer Networks”, arXiv:1506.02025 (2015).

Destination of shift



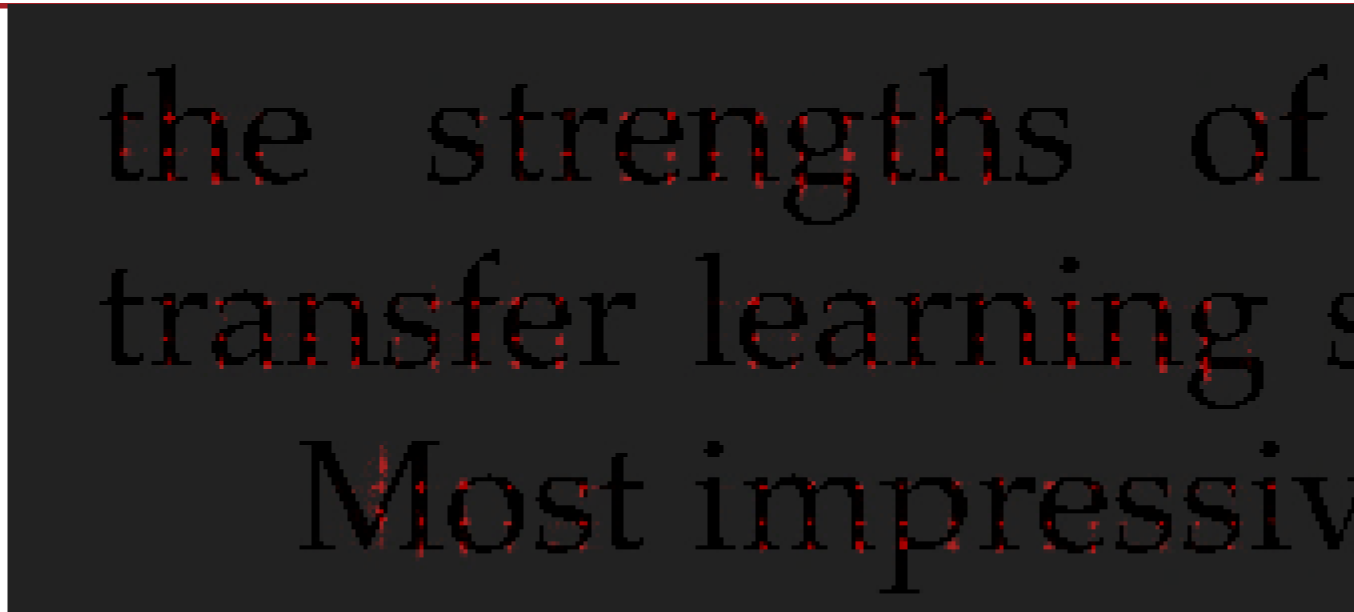
Above image represents a histogram of $\begin{bmatrix} x \\ y \end{bmatrix} + R\left(I_{x,y}\right)$,

where $I_{x,y}$ means a 32×32 patch extracted at $\begin{bmatrix} x \\ y \end{bmatrix}$.

(brighter red means more frequent.

We do not give information of each characters. However, histogram seems to highlights positions of typical patterns.

Destination of shift



Above image represents a histogram of $\begin{bmatrix} x \\ y \end{bmatrix} + R(I_{x,y})$,

where $I_{x,y}$ means a 32×32 patch extracted at $\begin{bmatrix} x \\ y \end{bmatrix}$.

(brighter red means more frequent.

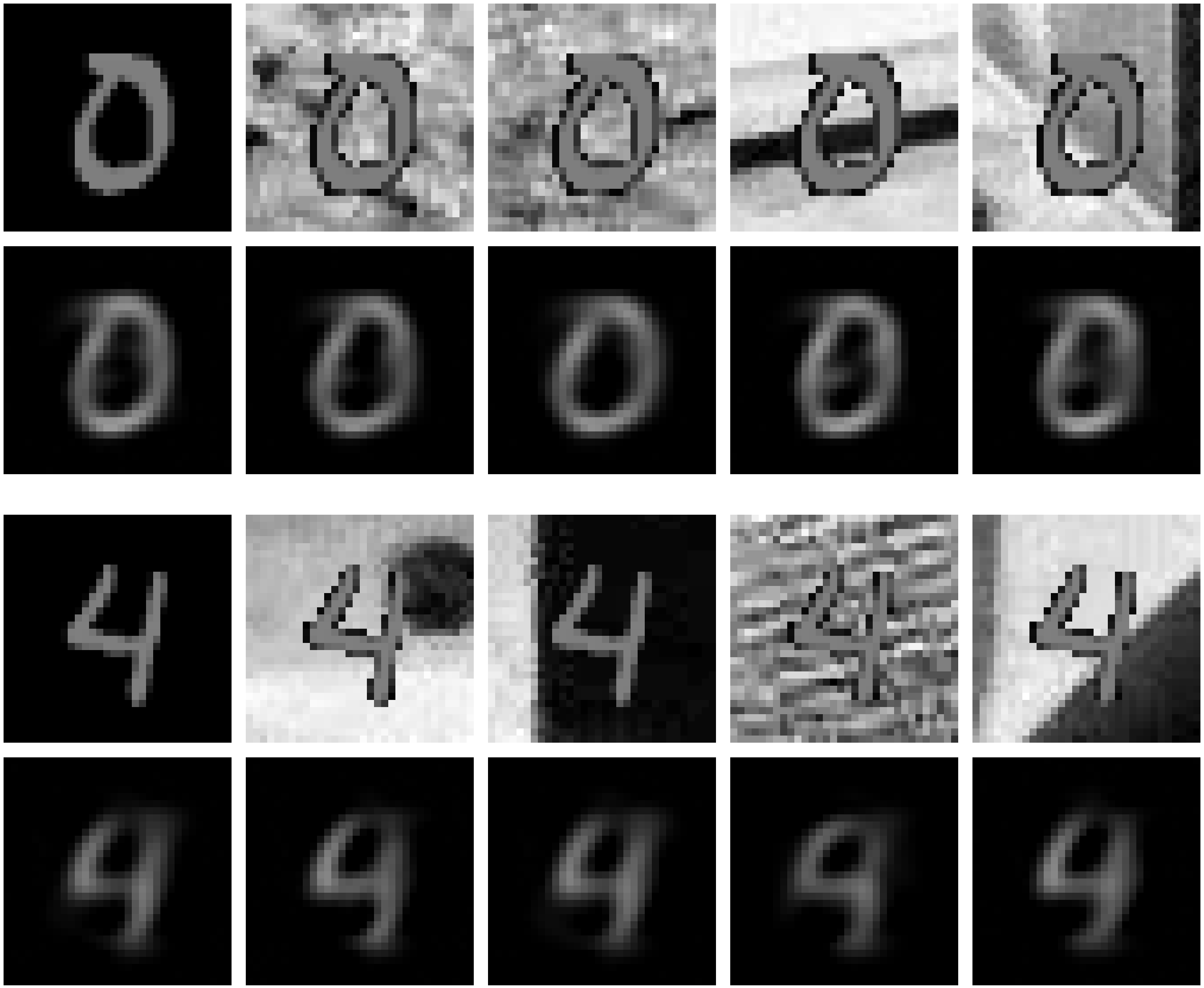
We do not give information of each characters. However, histogram seems to highlights positions of typical patterns.

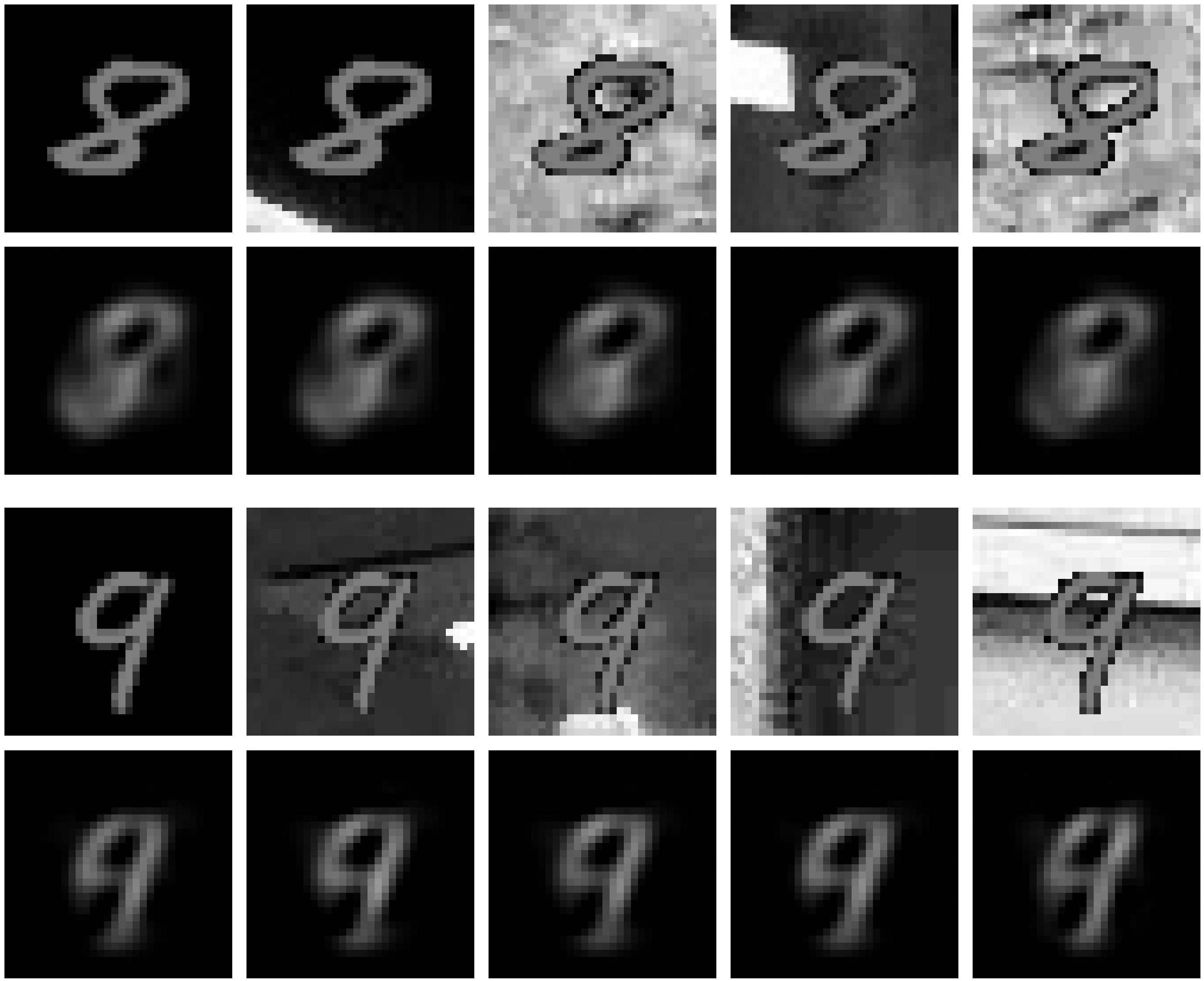
Input

Restored image

Input

Restored image





Input

Restored image

Input

Restored image

shift invariant and background alteration invariant auto-encoder