RITSUMEIKAN

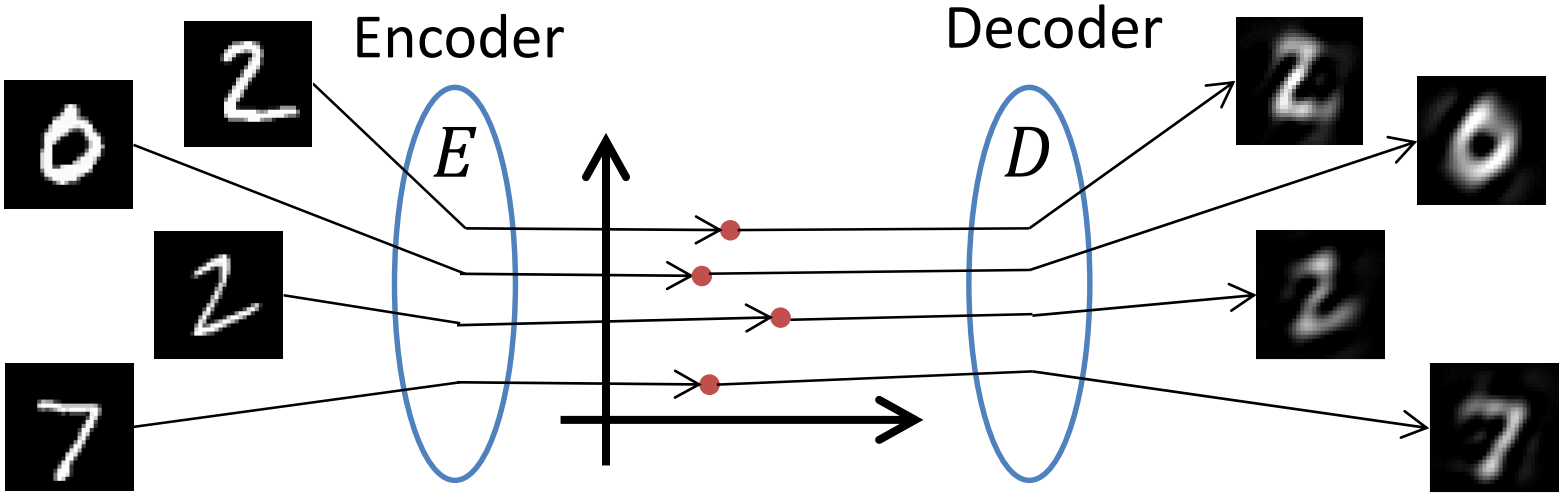
Segmentation based on Transform Invariant Auto-encoder

松尾 直志, 島田 伸敬
立命館大学

Beyond Borders

Auto-encoder

多くの入力について再現性を担保しつつ次元削減する方法



入力画像 I
32×32

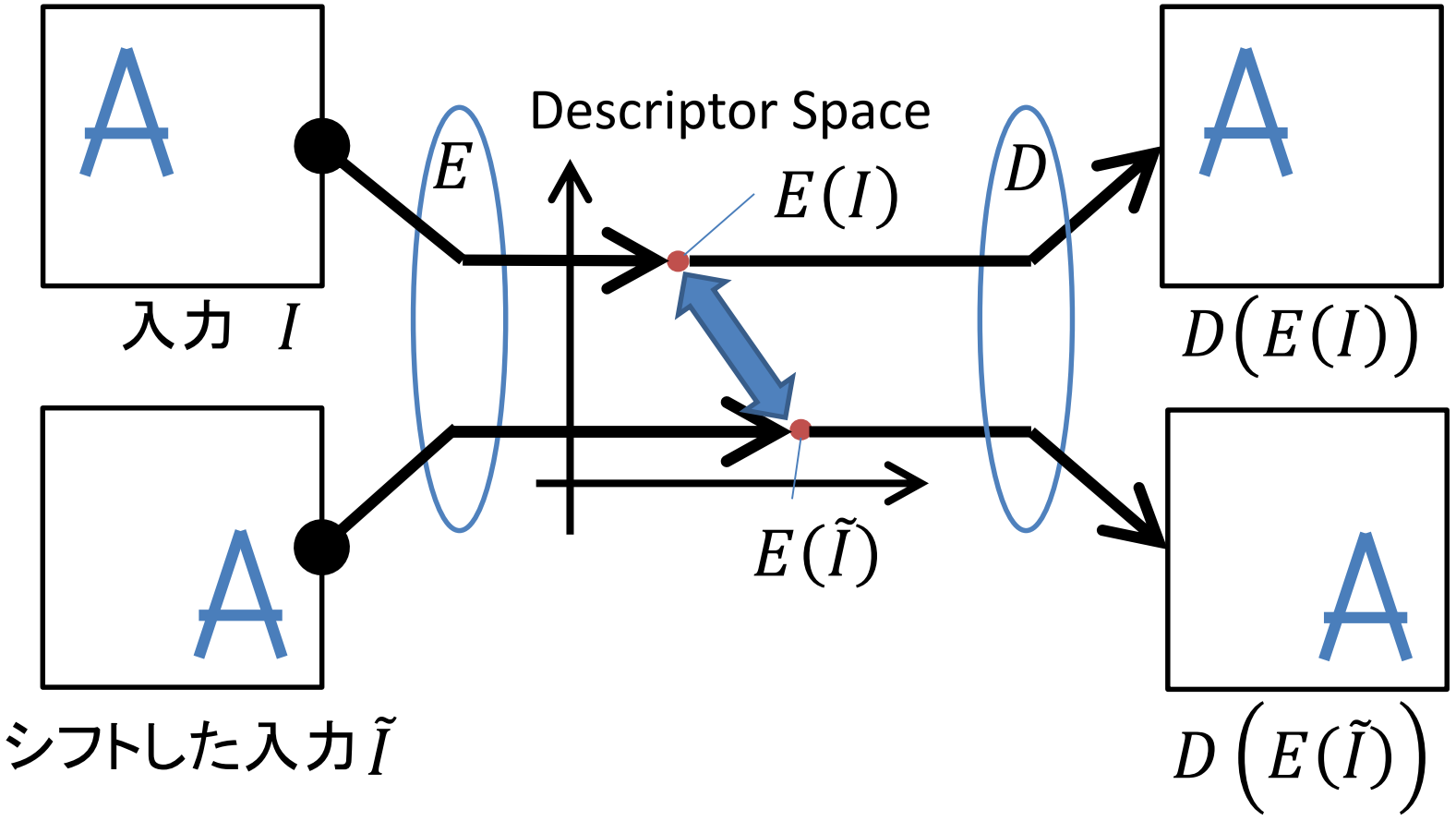
Descriptor $E(I)$
30

復元画像 $D(E(I))$
32×32

残差が最小となるようencoderとdecoderを学習する

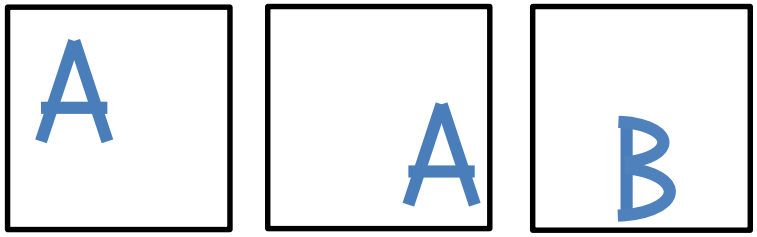
事前に教師ラベルを与えることなく、典型的なサンプルを
精度良く表現するdescriptorが得られる

通常のauto-encoder



同一形状であっても、その位置によってdescriptorが異なる

通常のauto-encoderの問題



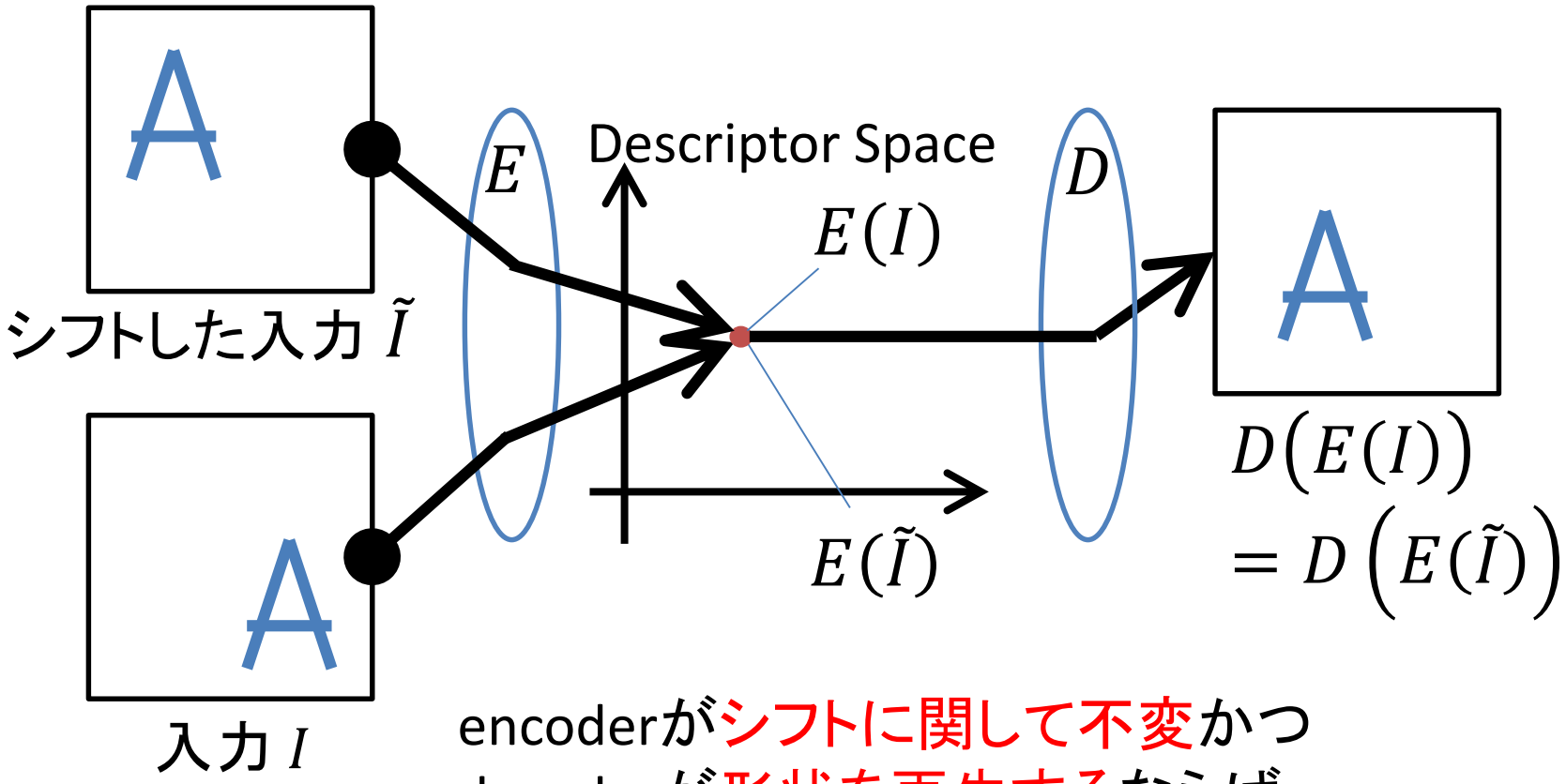
通常のauto-encoderでは左の画像のdescriptorはすべて異なる
形状を比較するには要正規化

正規化が難しい
対象もある

事前に明示的な正規化をすることなく、
同じ形状の画像は同一descriptorとなるようにできないか？

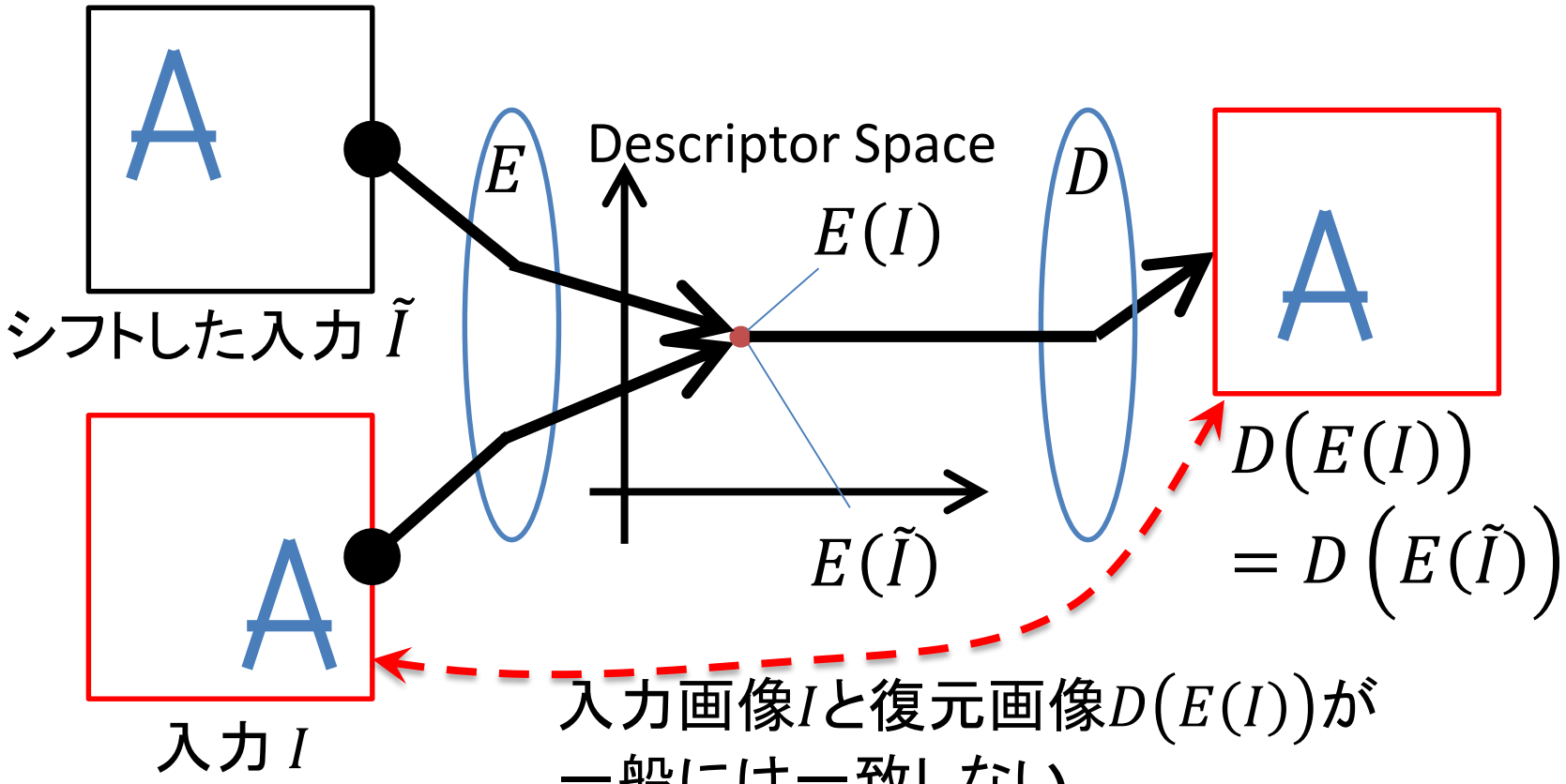
auto-encoderがシフトに関して不変で
あれば、形状そのものを表すdescriptorが得られる

シフト不変auto-encoder



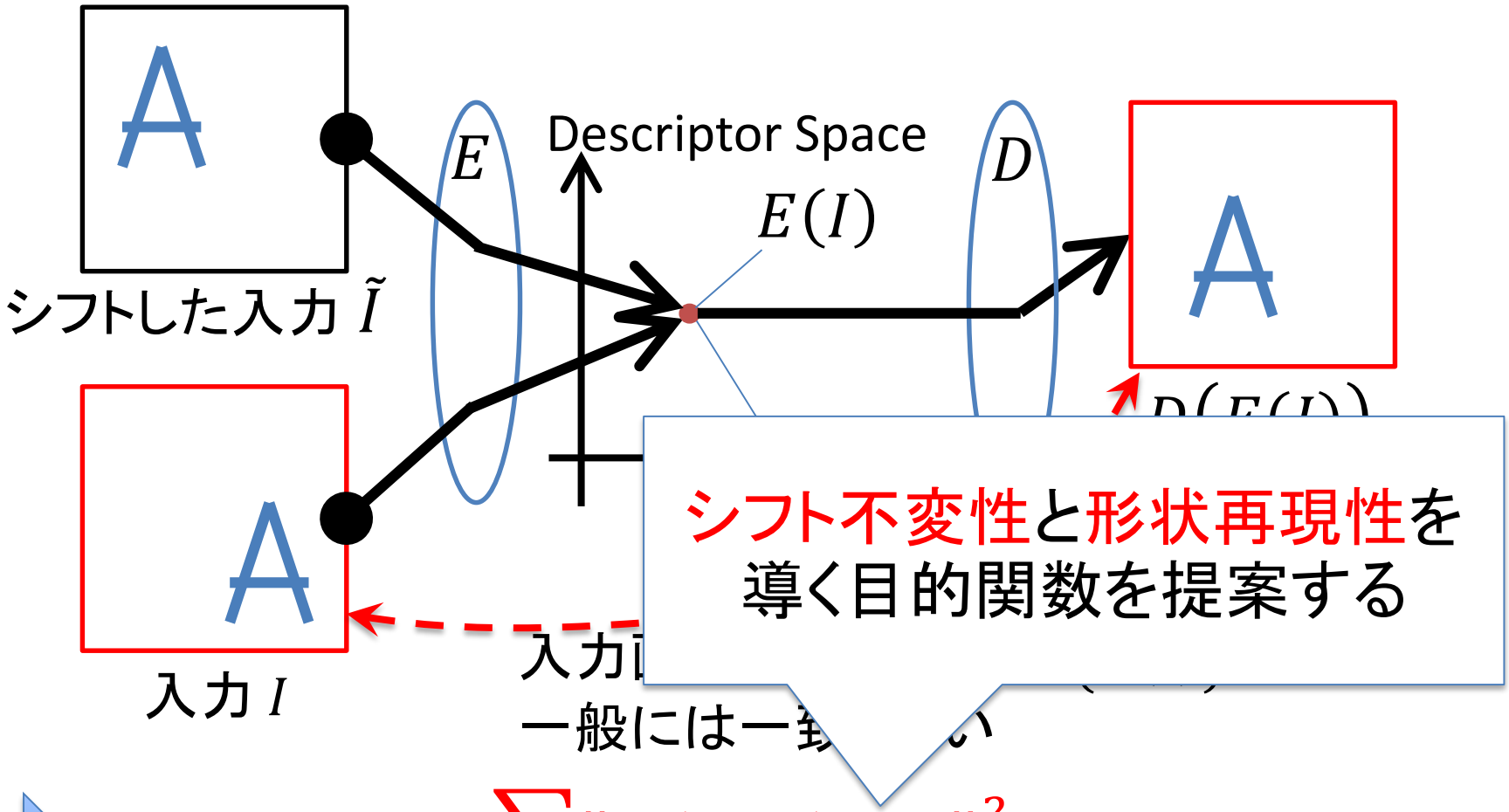
encoderがシフトに関して不変かつ
decoderが形状を再生するならば
形状自体を表すdescriptorが得られる

シフト不変auto-encoder



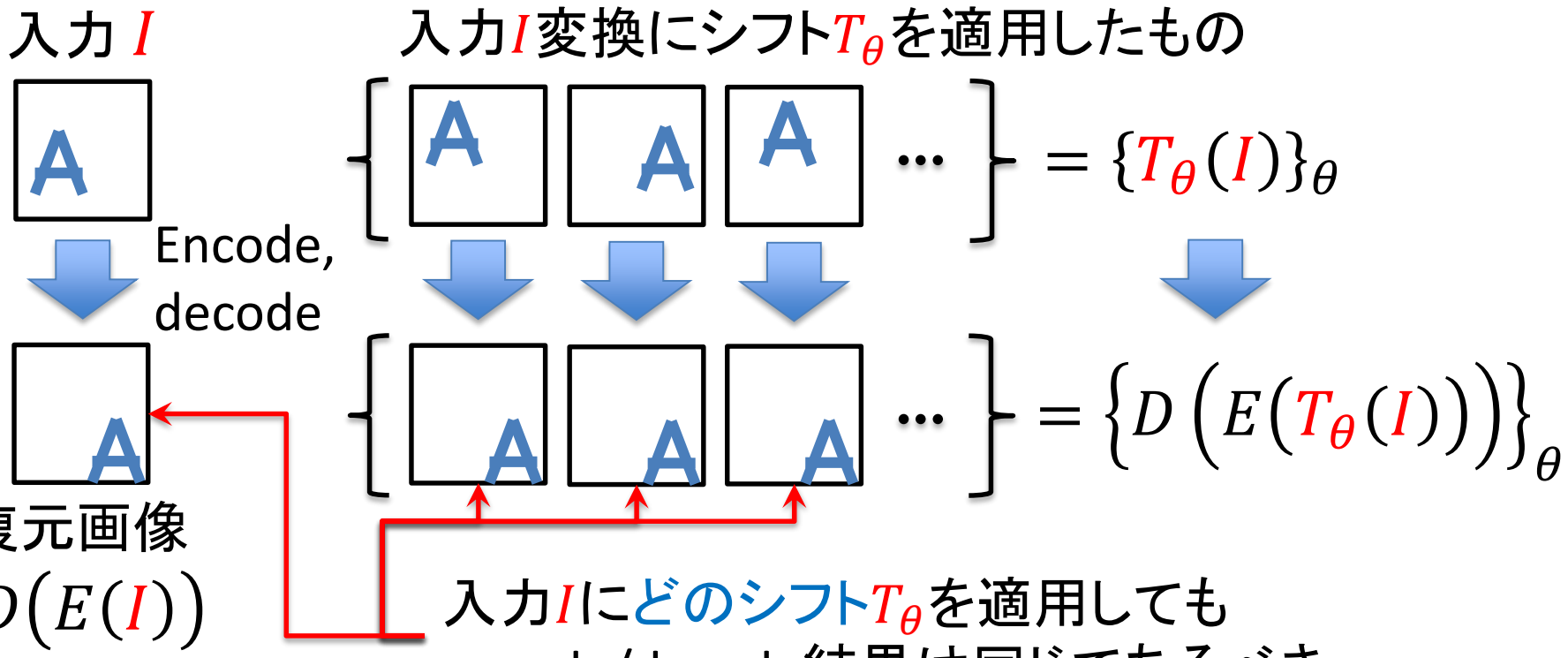
通常の目的関数 $\sum_I \|D(E(I)) - I\|_{L2}^2$ では学習できない

シフト不変auto-encoder



通常の目的関数 $\sum_I \left\| D\left(E(I)\right) - I \right\|_{L2}^2$ では学習できない

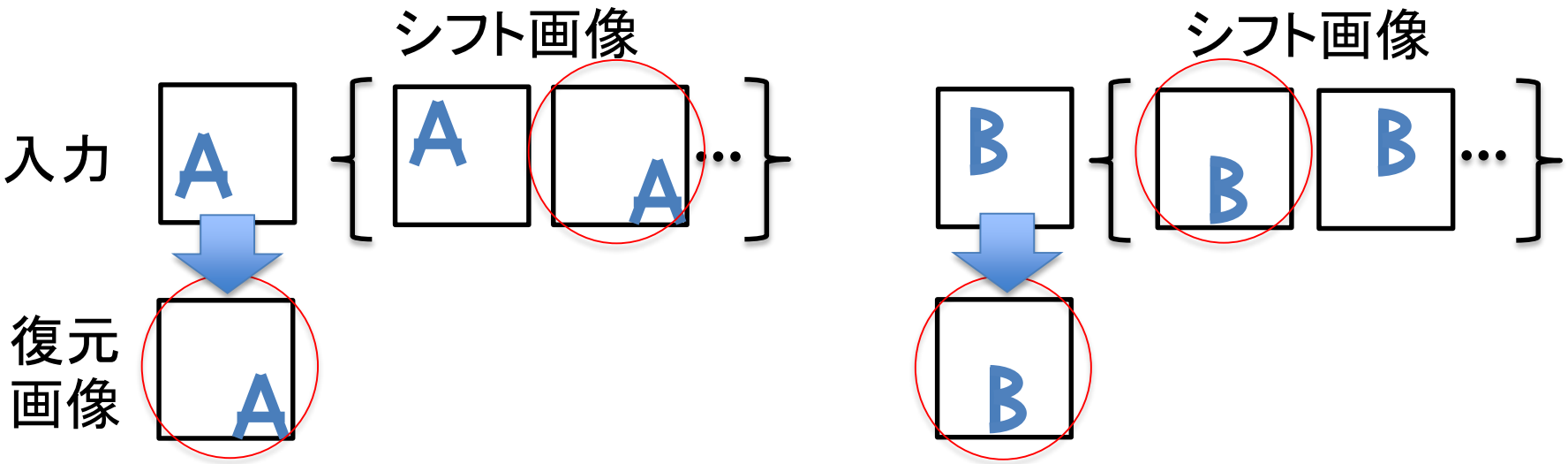
シフト不変性の評価



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

これが小さくなるようにE, Dを学習する

再現性の評価



入力をシフトしたもののいずれかと復元画像が
一致するなら形状が保存されているといえる

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

これが小さくなるように
E, Dを学習する

復元画像に最も近い画像を与える変換を採用する

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

提案目的関数

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

不変性項 $C_{inv}(E,D)$

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

入力Iをいずれの変換T_θで変換しても
encode/decode結果が変わらない制約

再生性項 $C_{res}(E,D)$

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

encode/decode結果は入力Iをいずれかの
変換で変換したものに近いという制約

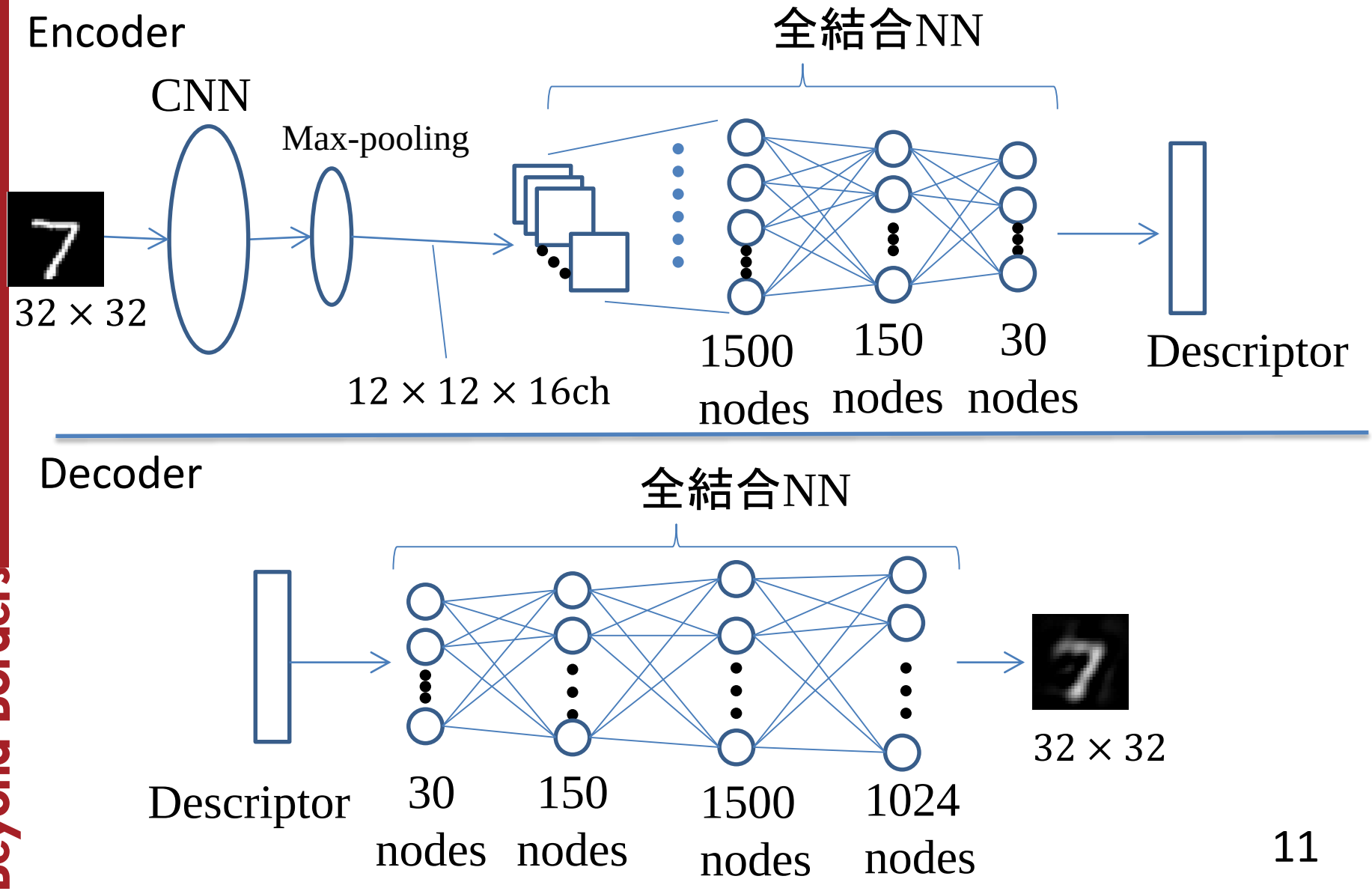
Sparseness項

$$C_{spa}(E)$$

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

descriptor $E(I)$の
エネルギーが少数
の次元に集中して
いるべきという制約

Auto-encoderの構造



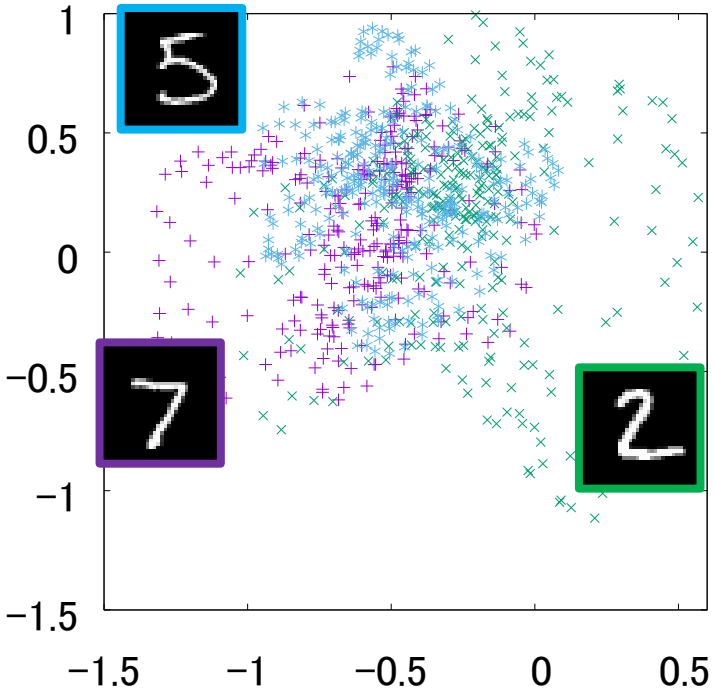
シフト不変auto-encoderの例

MNISTの訓練画像をシフトしたものを学習

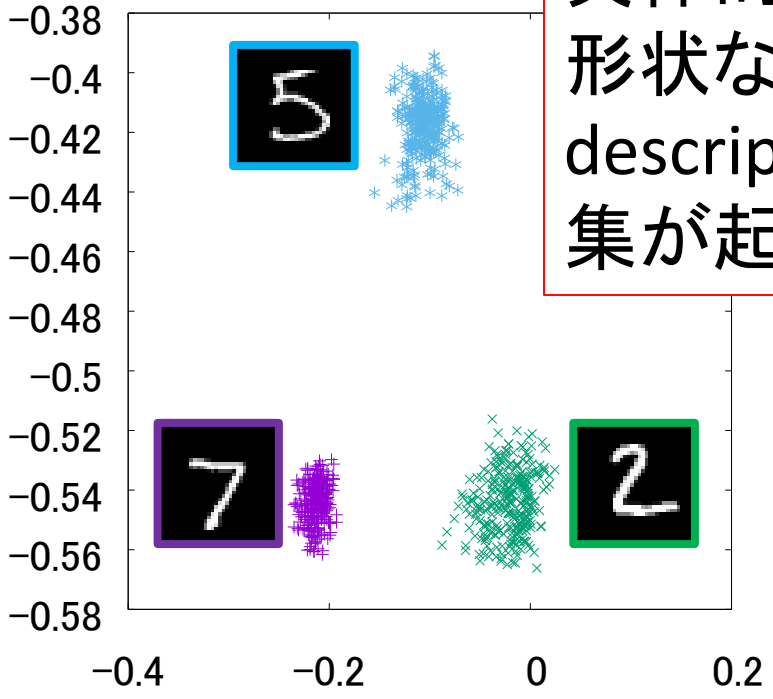
入力:32 × 32
Descriptor: 30次元
最大シフト幅: 8

テスト画像をシフトして
descriptorを計算

具体的な基準形状なしにdescriptorの凝集が起こる



通常のauto-encoder

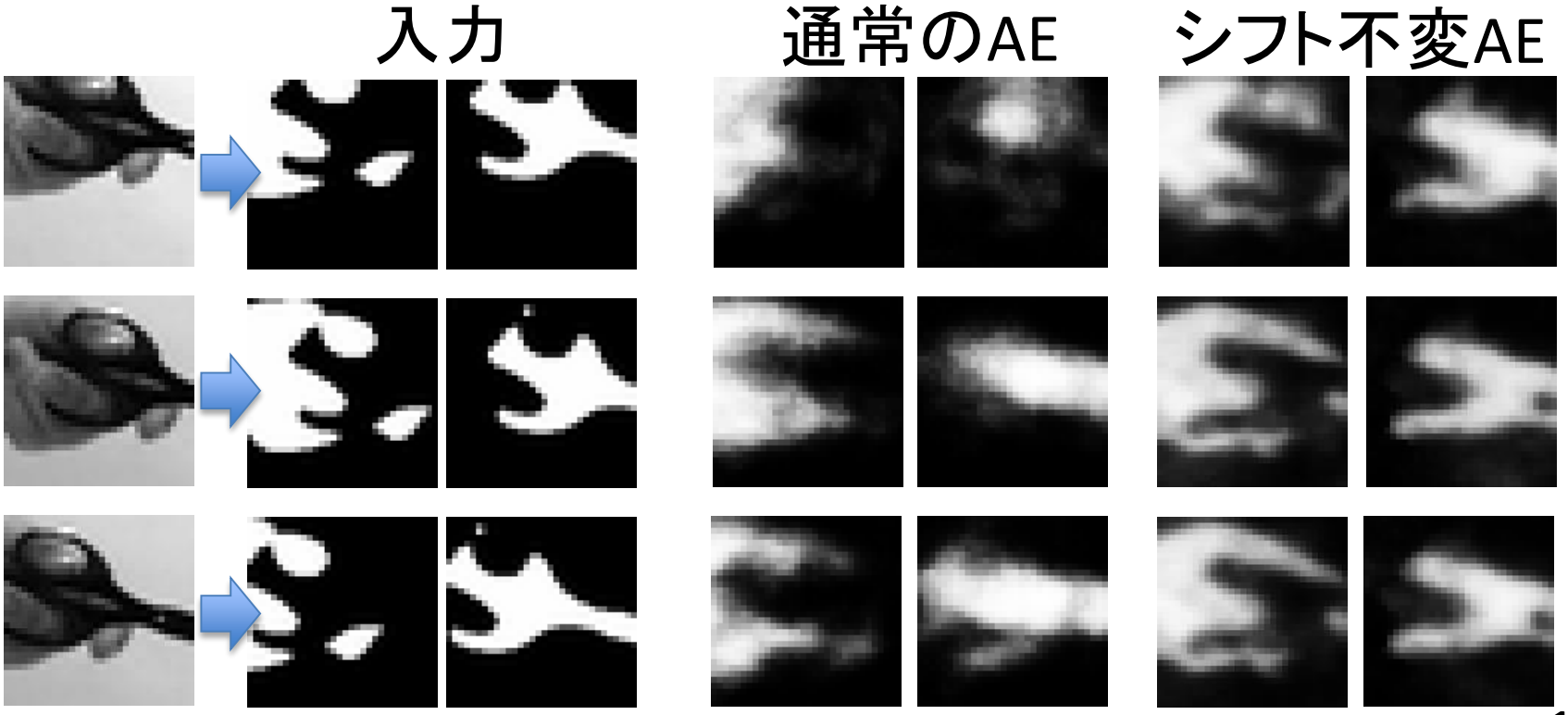


シフト不変auto-encoder

物体把持画像への適用例

物体把持時の手領域マスクと物体領域マスクの組を学習

encode/decodeした復元画像



物体把持画像への適用例

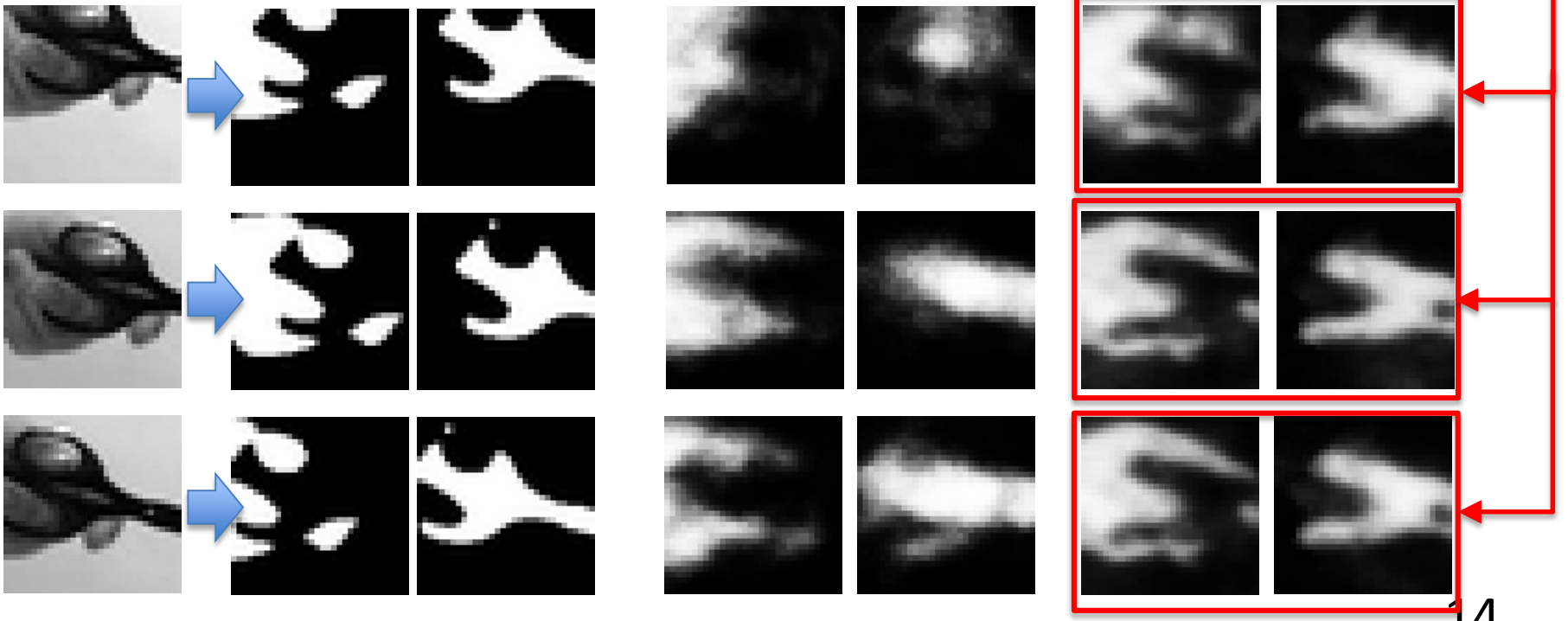
手領域　物体領域

物体把持時
体領域マス

ハサミを握っている
状況を表すdescriptor

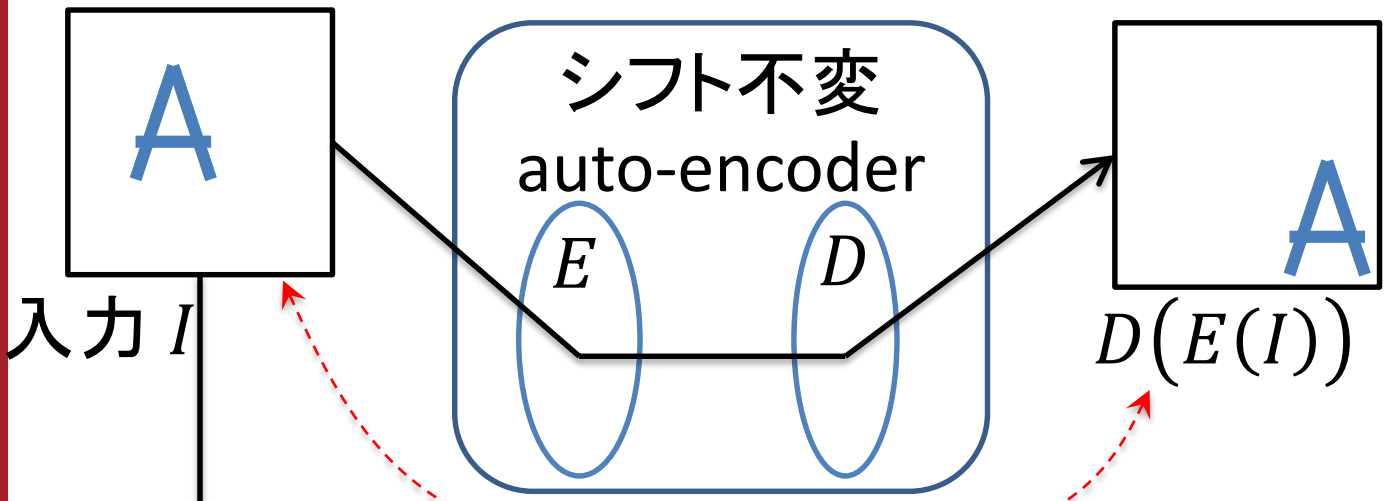
encode/decodeした復元画像

入力　　通常のAE　　シフト不変AE

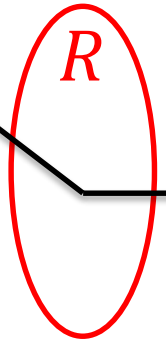


Beyond Borders

シフト量推定



入力と復元画像を比較して
最適なシフトが求まる



シフト量
推定器

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

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

シフト不変性によって画像を形状と位置の2つに分割

シフト不変auto-encoderによる形状と位置への分解の例

入力

復元画像

形状を復元後、推定した量だけシフトした画像

$D(E(I))$

形状のみ

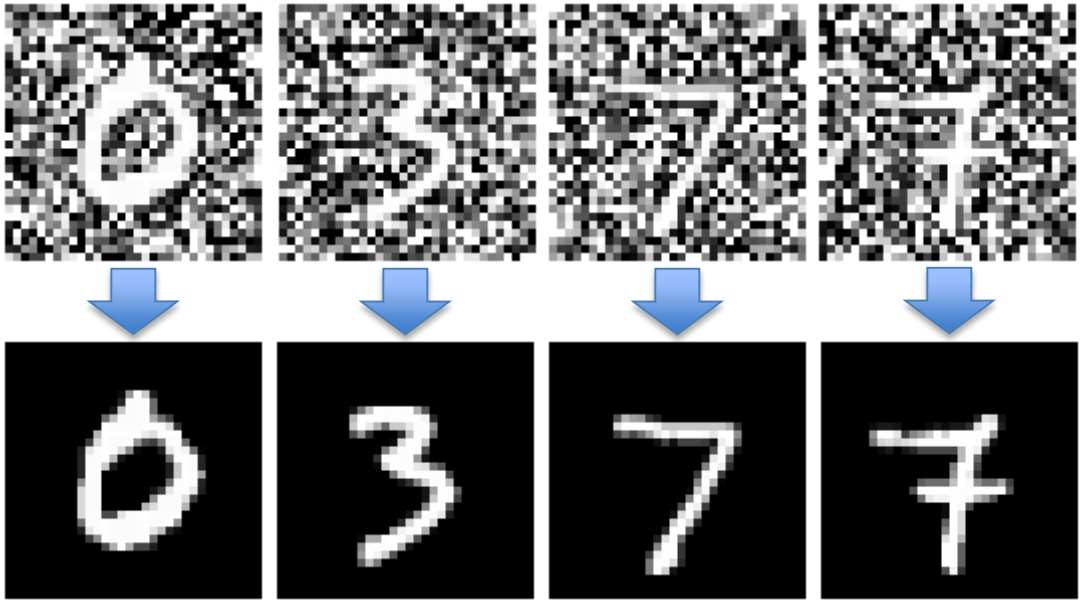
$T_{R(I)}\left(D(E(I))\right)$

形状, 位置情報

入力Iを形状を表すdescriptor $E(I)$ と
シフト量を表す $R(I)$ に分解できる

Beyond Borders

複雑背景を持つ画像からのパターン抽出への適用

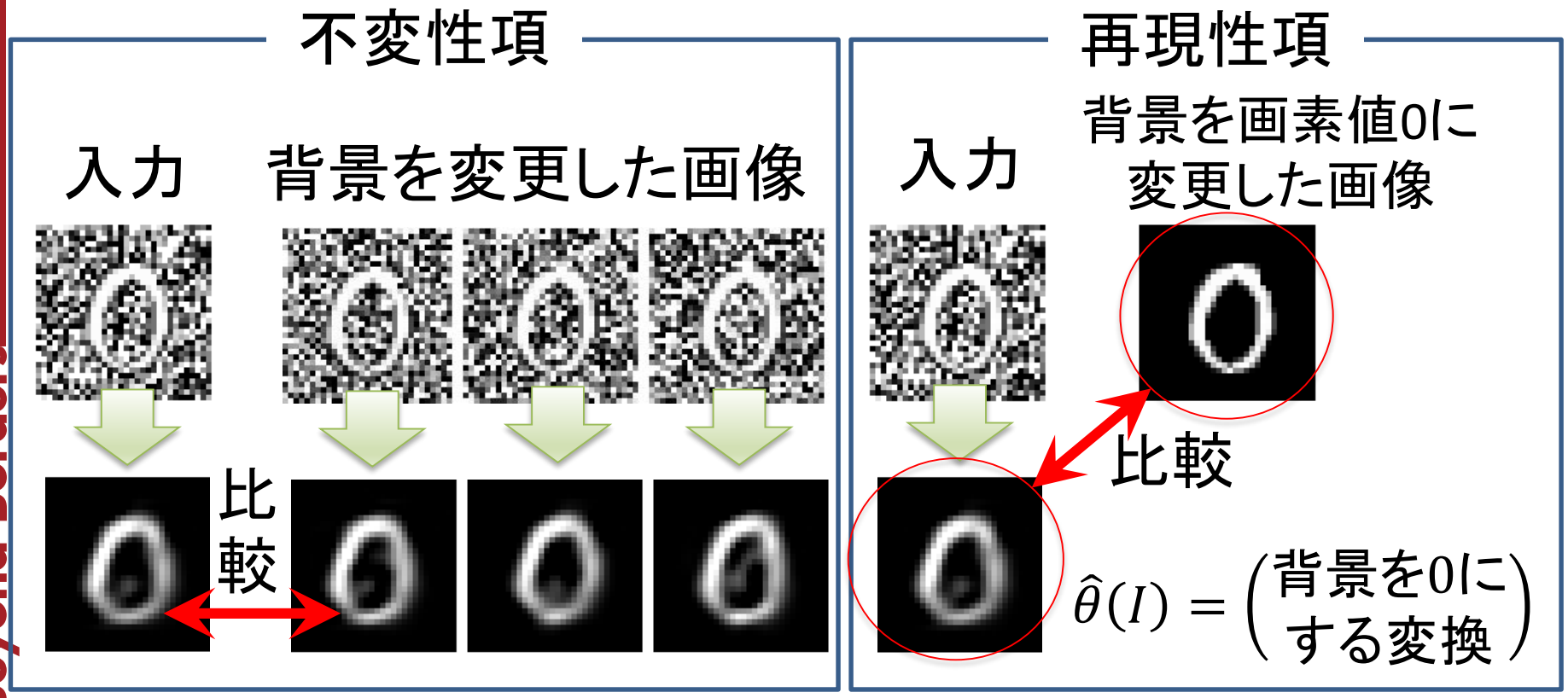


手書き数字のような多くのバリエーションのあるパターンを複雑な背景を持つ画像から抽出する問題に利用できる

前景抽出auto-encoder

背景を変更する変換に関して不変なauto-encoderを構成

前景の文字部分（背景に依存しない成分）を
表現するdescriptorが得られる



Beyond Borders

前景抽出auto-encoderの目的関数

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

不変性項 $C_{inv}(E,D)$

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

入力Iを変換T_θで変換してもencode/decode結果が変わらない制約

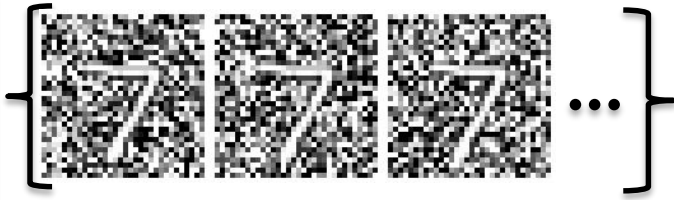
再生性項 $C_{res}(E,D)$

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

encode/decode結果は入力Iの背景を0にしたものに近いという制約

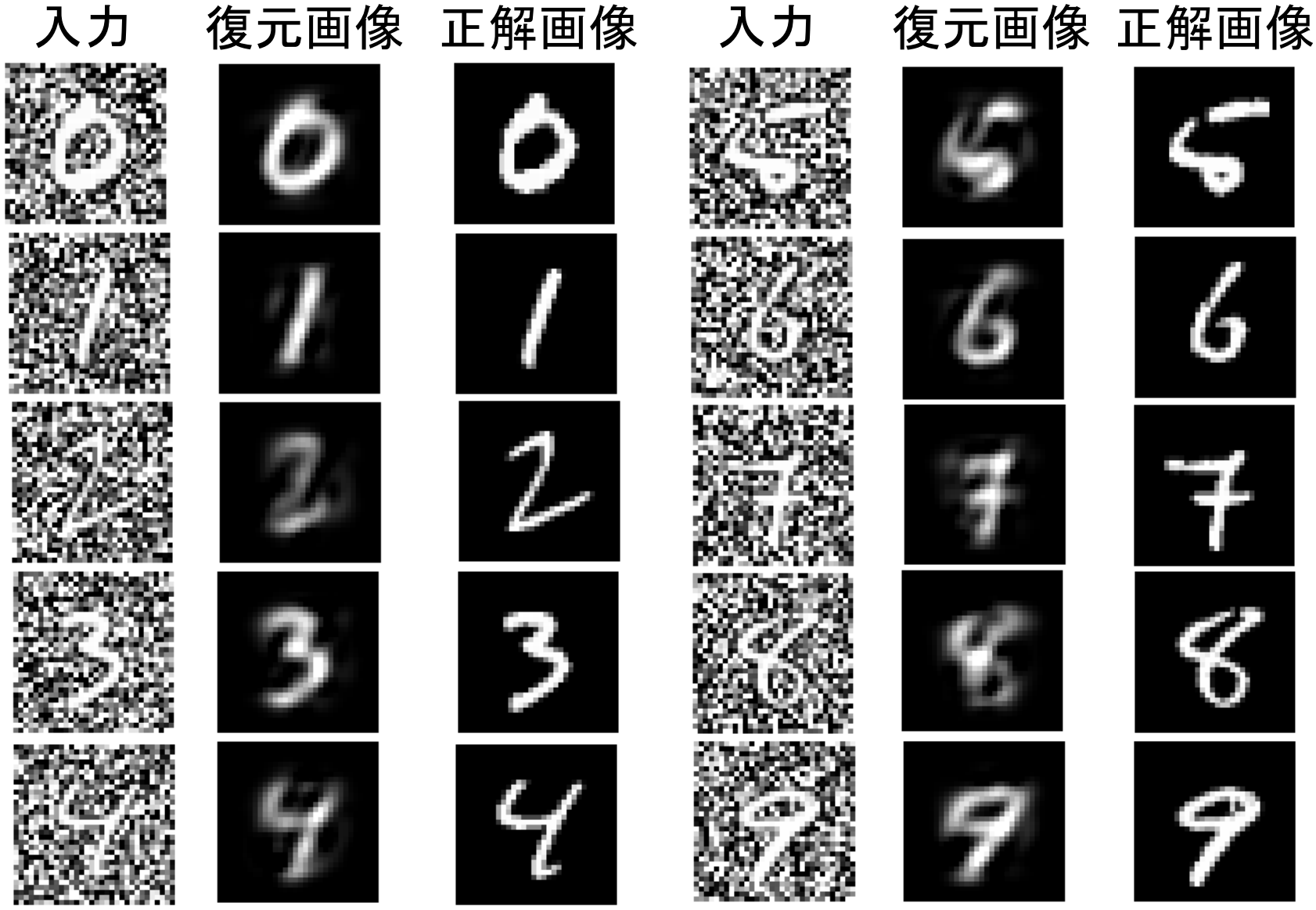
入力 I

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

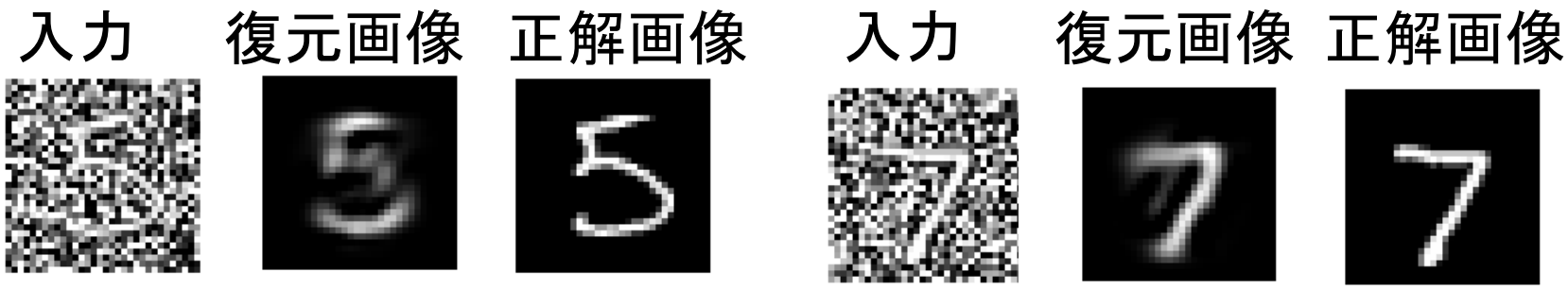


$T_{\hat{\theta}(I)}(I) =$

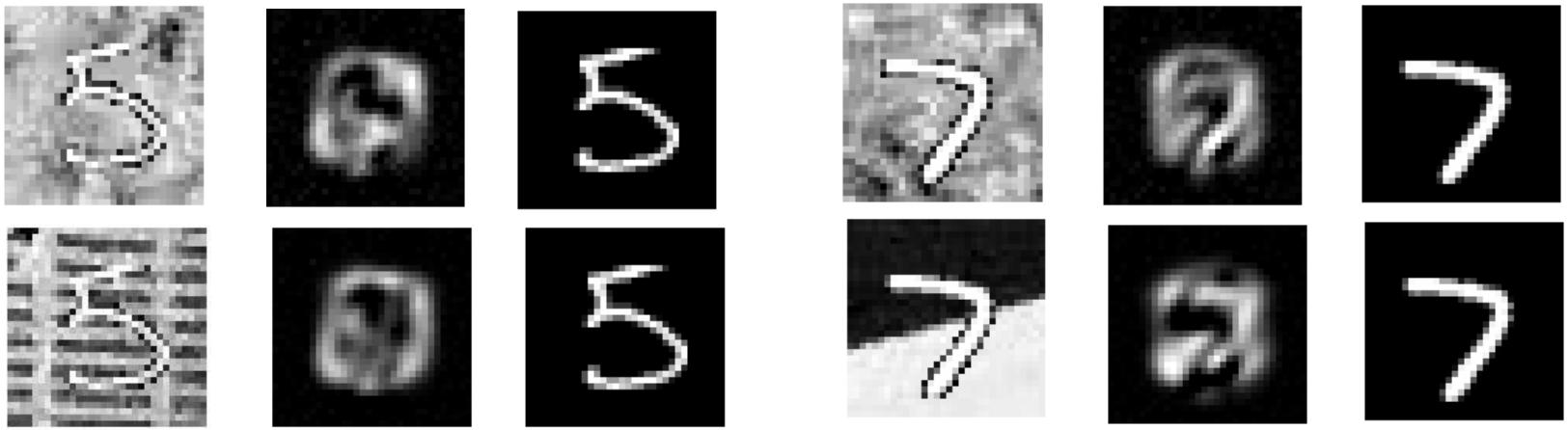
雑音背景からの復元例



背景への依存性

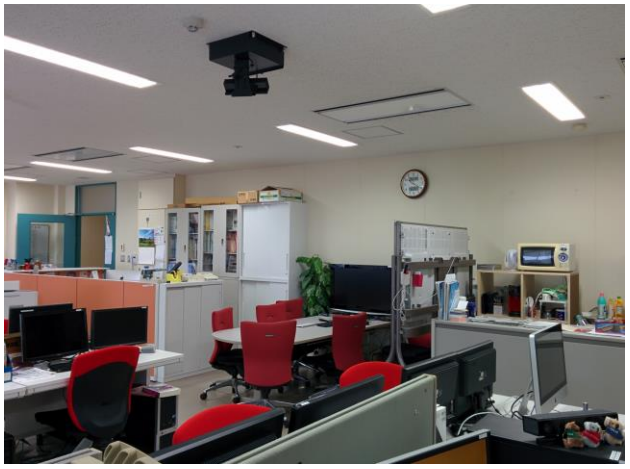


室内風景を背景に持つ画像に適用した場合



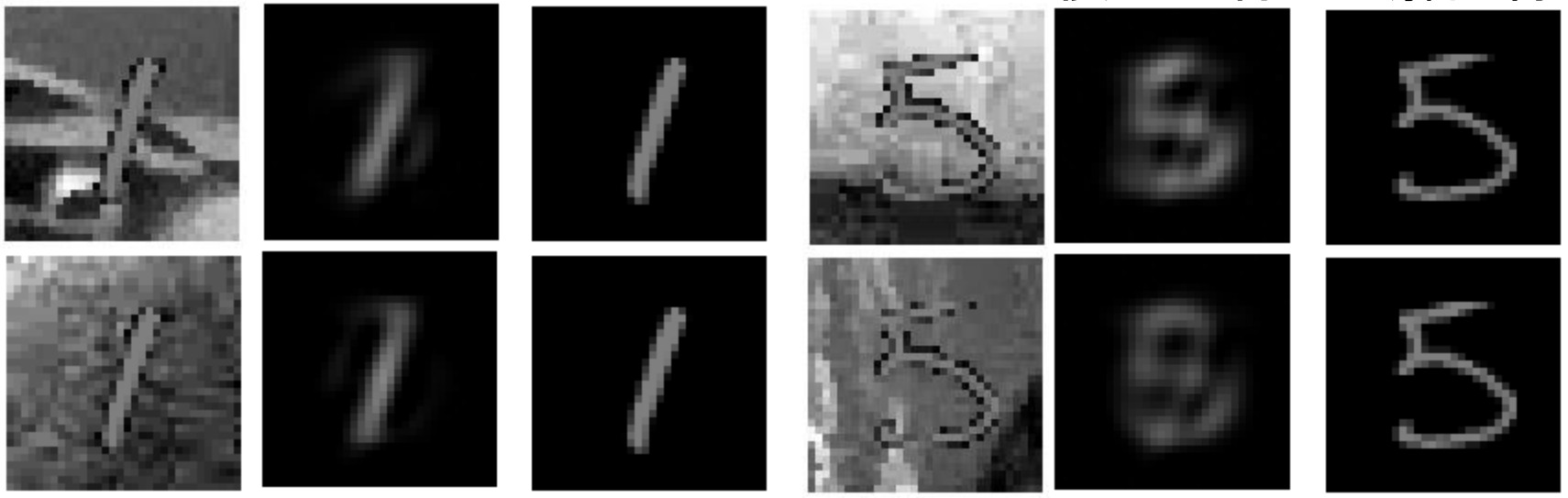
雑音背景で学習したauto-encoderでは、室内風景を背景に持つ画像からの抽出はできない

室内風景背景からの抽出例

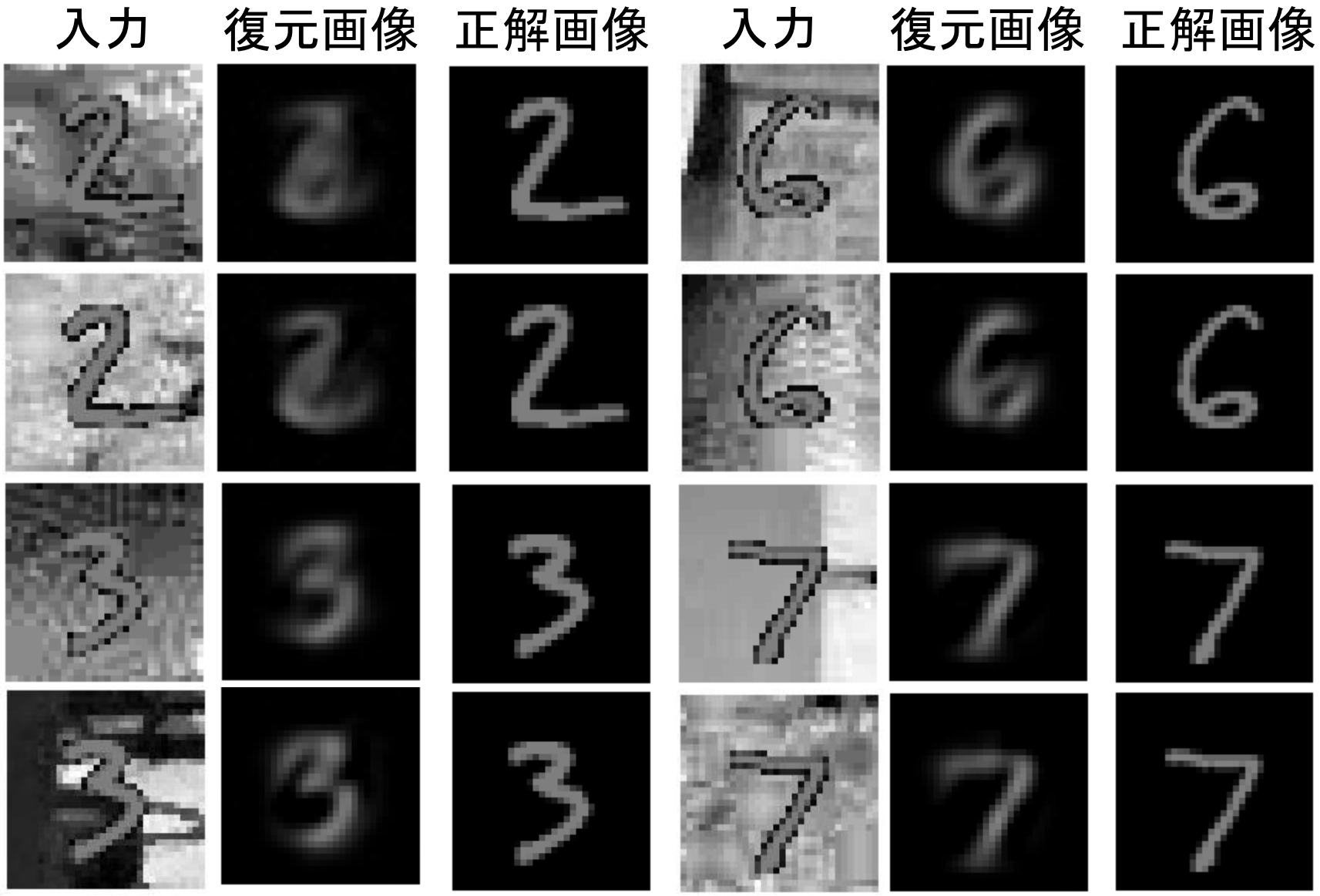


3200×2368の室内写真から切り出した32×32を背景として学習

入力　復元画像　正解画像　入力　復元画像　正解画像



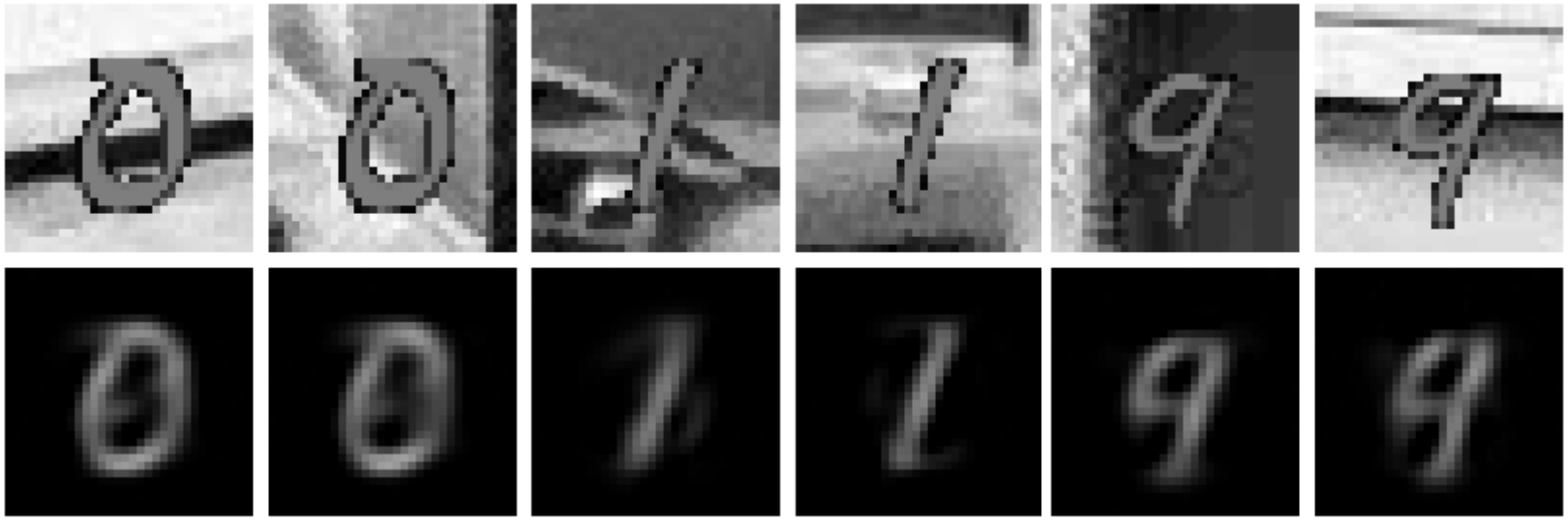
室内背景からの抽出例



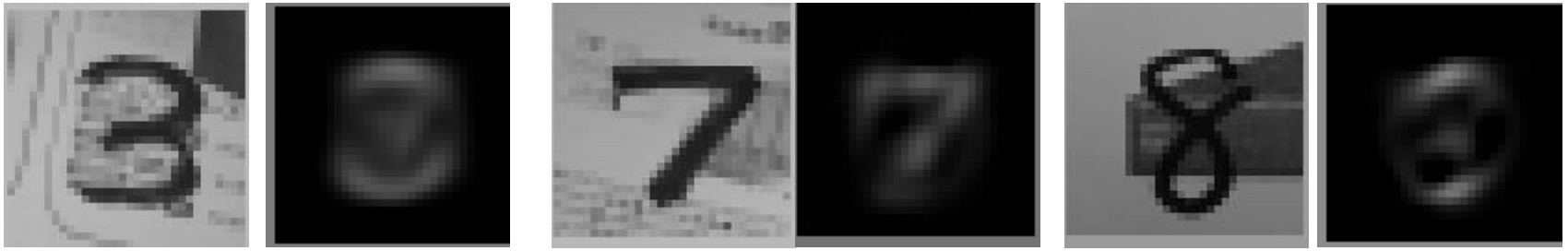
室内背景からの抽出例

入力

復元画像



チラシ上の手書き文字



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

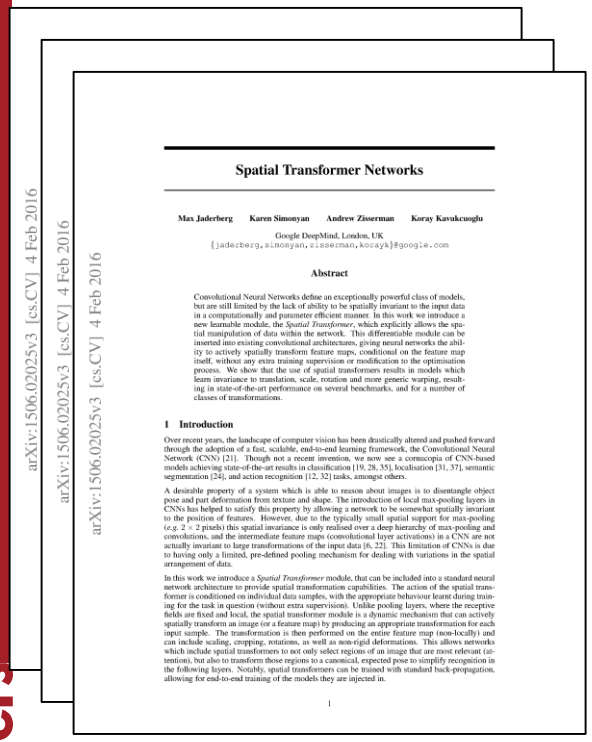
前景は不変

- 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

文書画像への適用例



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.

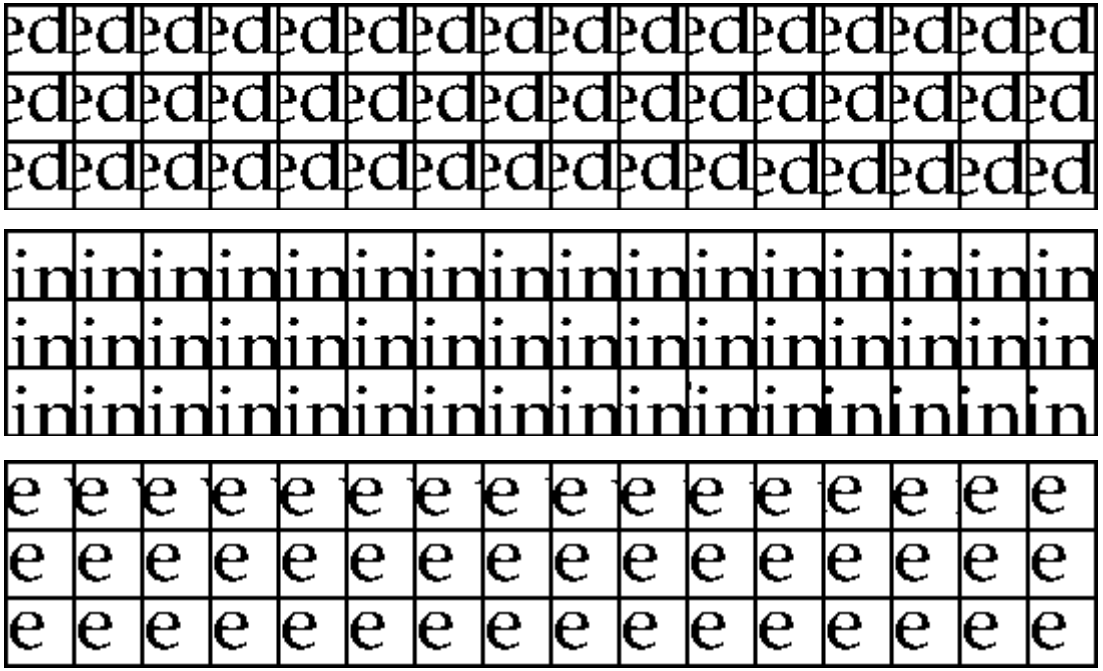
In this work we introduce a *Spatial Transformer* module, that can be included into a standard neural network architecture to provide spatial transformation capabilities. The action of the spatial transformer is conditioned on individual data samples, with the appropriate behaviour learnt during training 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

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

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

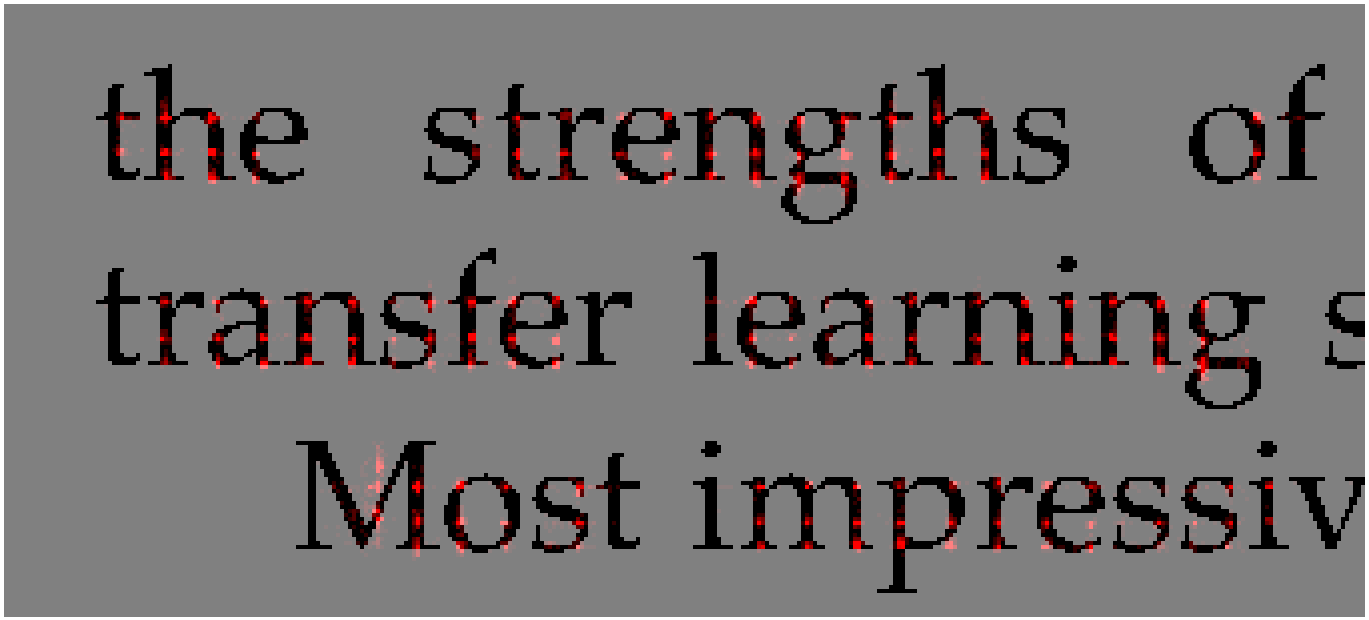
arXiv:1506.02025v3 [cs.CV] 4 Feb 2016



Descriptorに対してmean shift clusteringを適用すると文字の組に対応するclusterが得られた

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

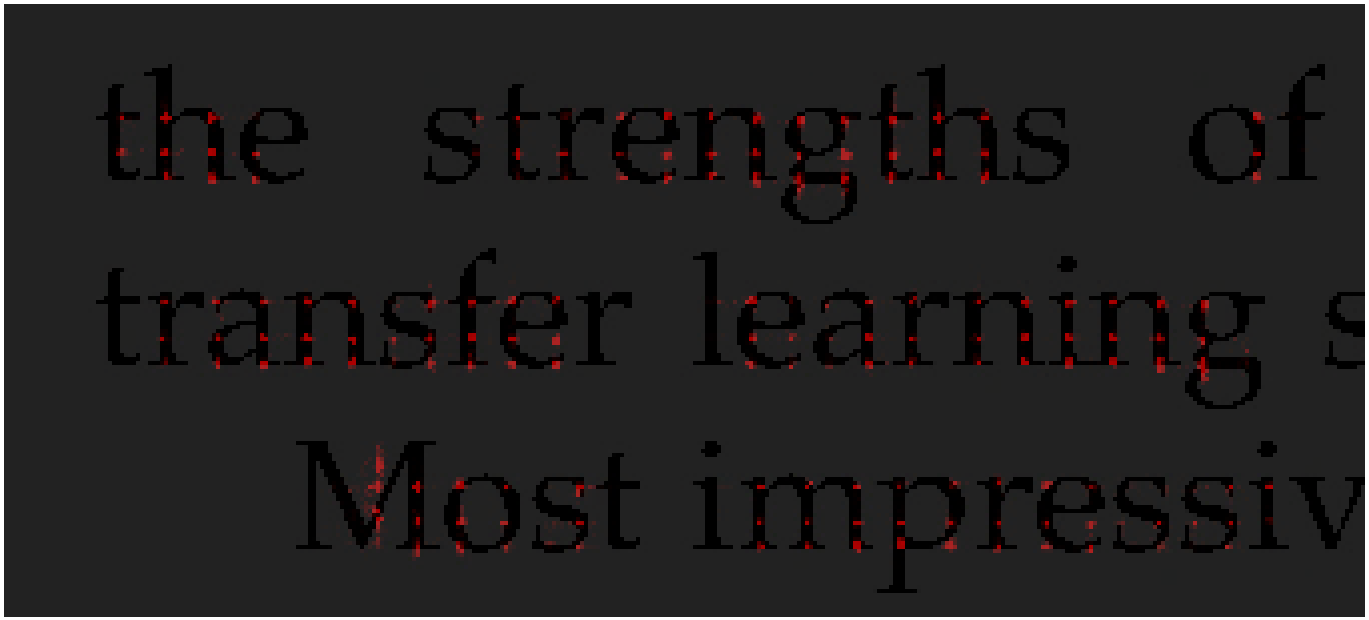
シフト量推定



文書中の各場所から32x32のパッチを作成して
シフト量推定を行い、そのシフト先となった頻度を図示
（赤いほど高頻度）

文字についての情報は与えていないにも関わらず、
典型的なパターンのある場所が発見されている

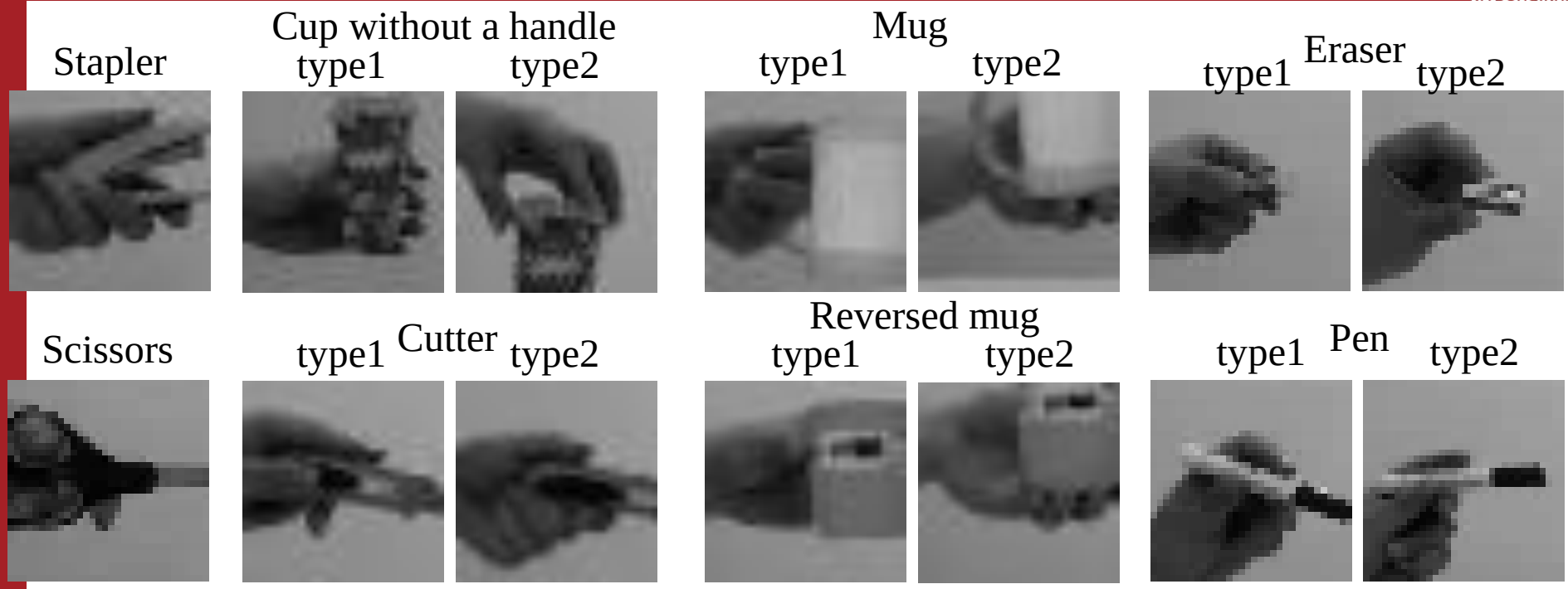
シフト量推定

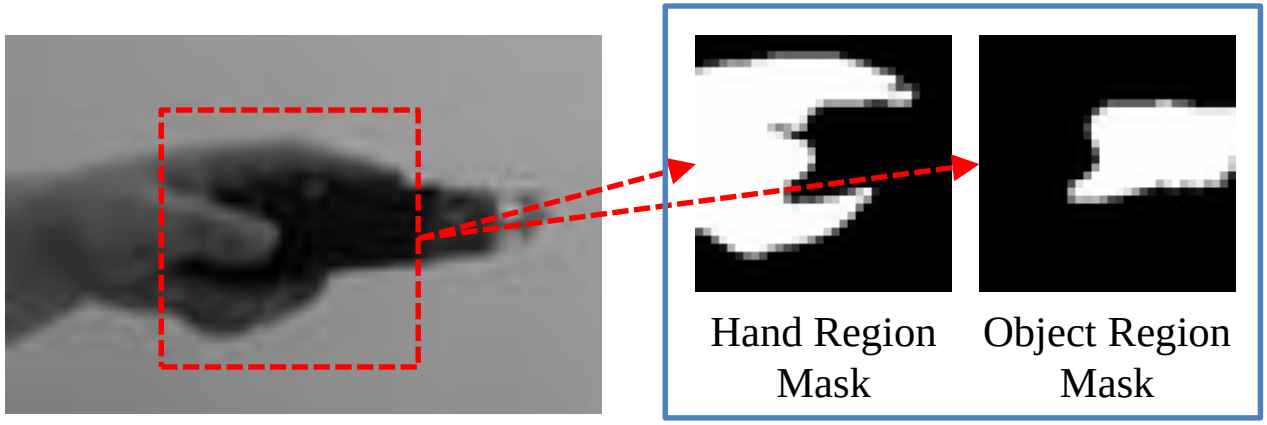


文書中の各場所から32x32のパッチを作成して
シフト量推定を行い、そのシフト先となった頻度を図示
（赤いほど高頻度）

文字についての情報は与えていないにも関わらず、
典型的なパターンのある場所が発見されている

学習に使用した把持パターン

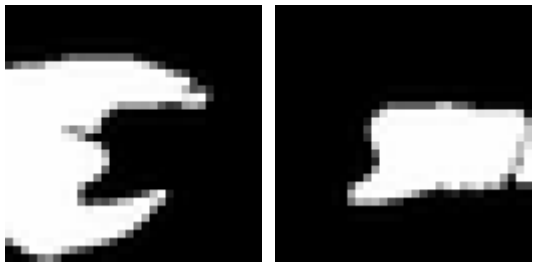
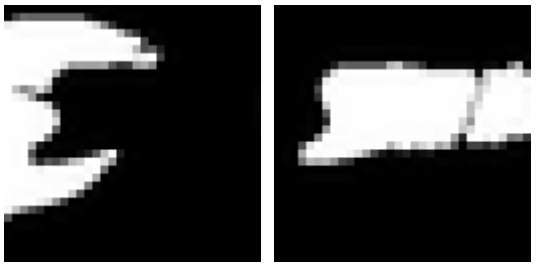
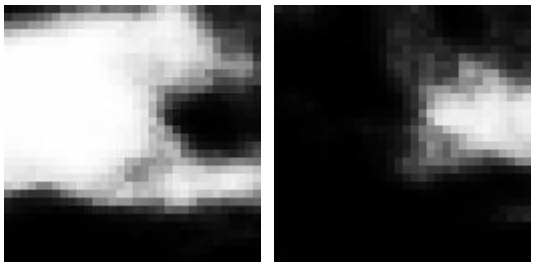
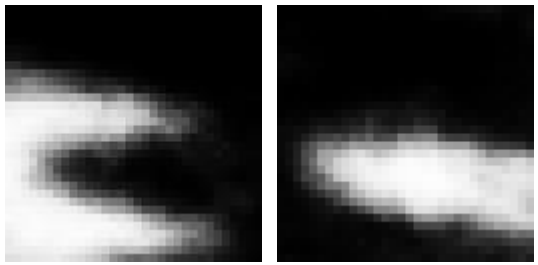
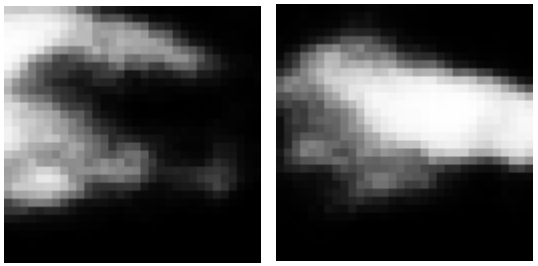
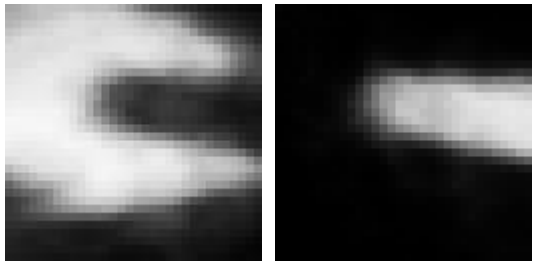
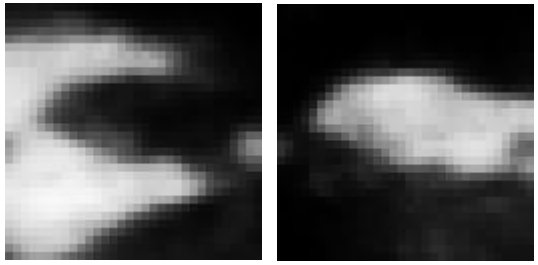
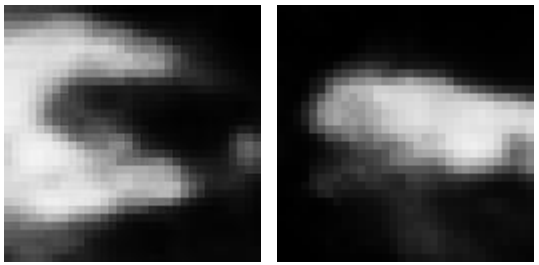


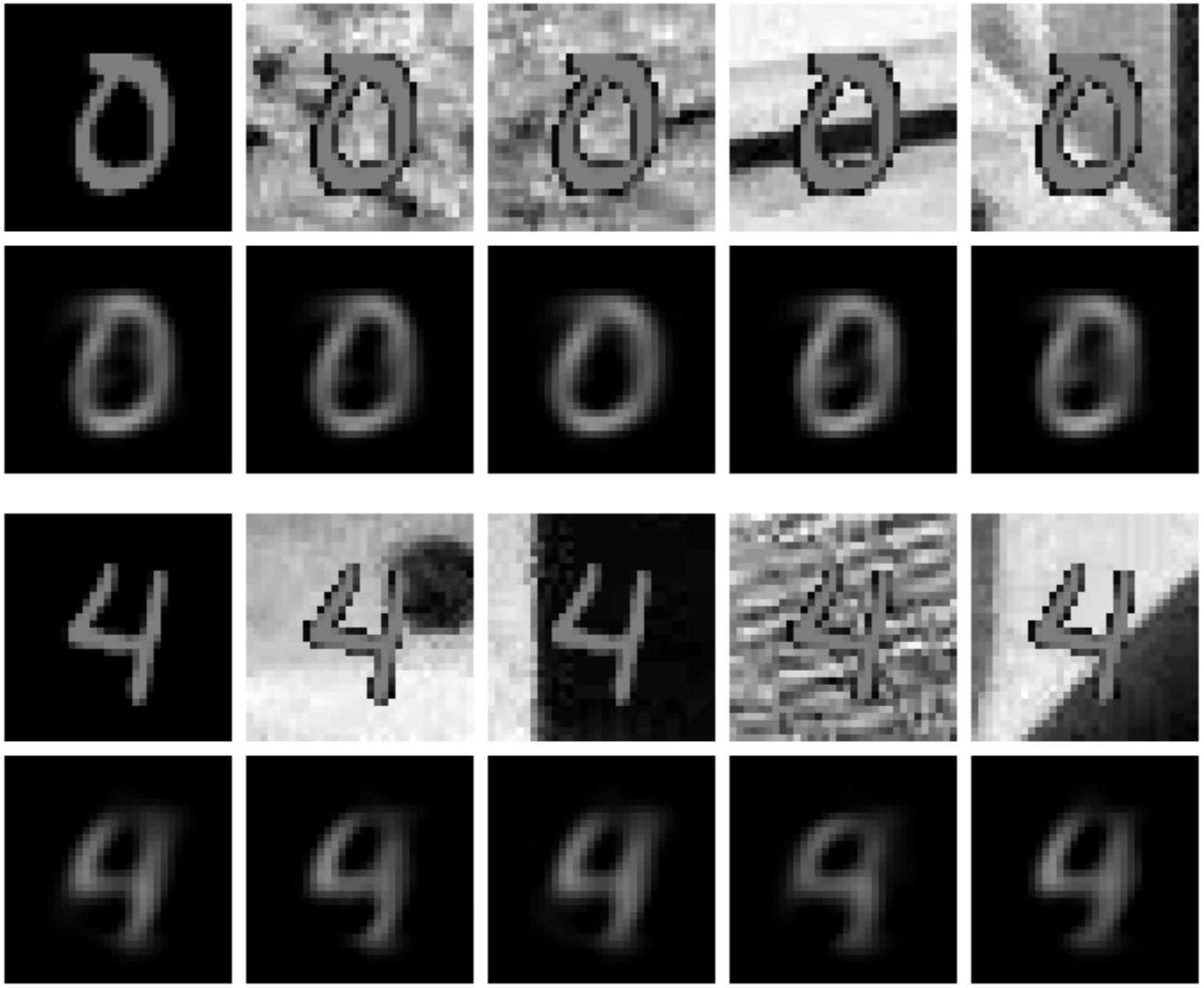


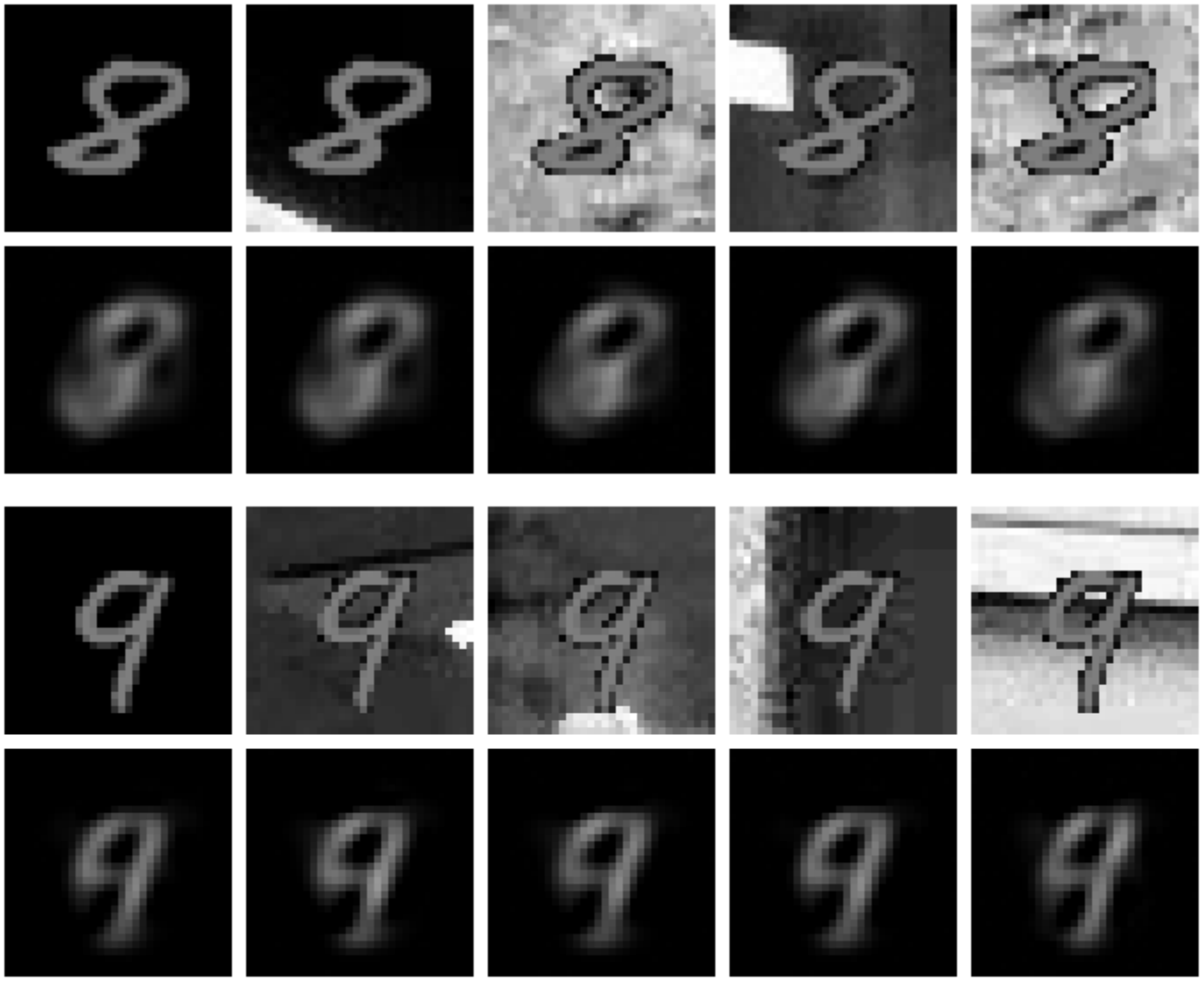
Interaction Image (32*32*2ch)



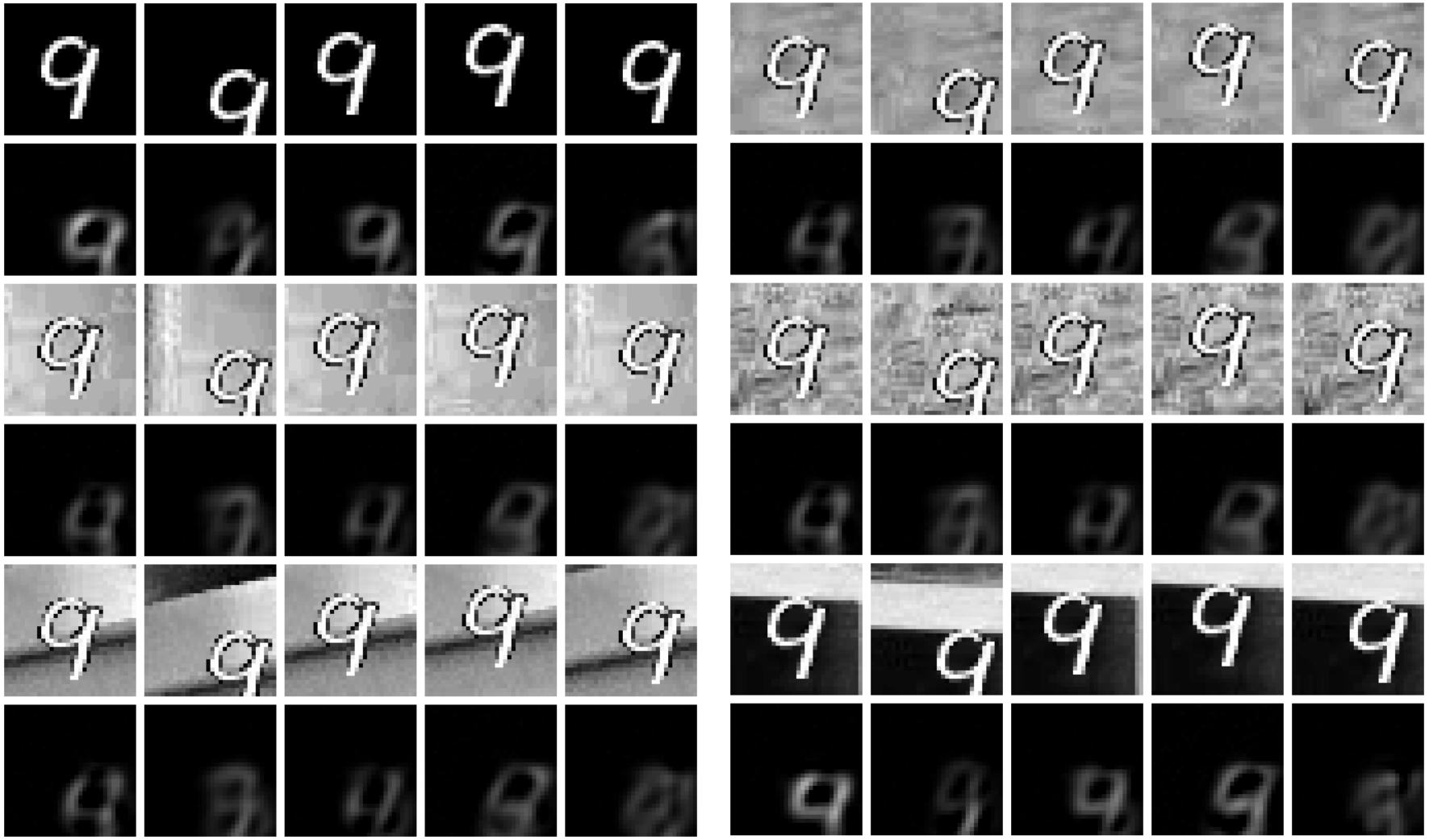




Input image I Input image I Input image I



Images restored by an ordinary auto-encoder



Images restored by a shift invariant auto-encoder

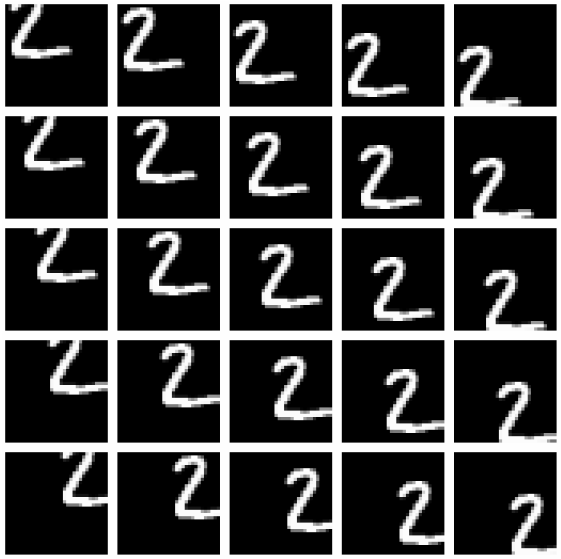




シフトと背景変換について不変なauto-encoderによる復元の例

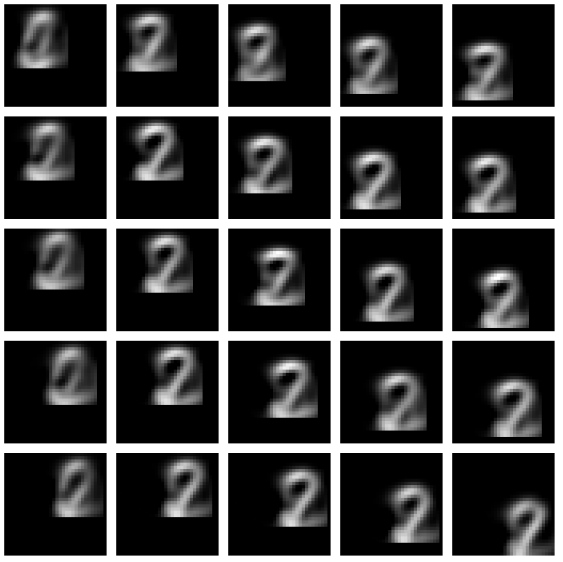


シフト不変auto-encoderによる復元画像



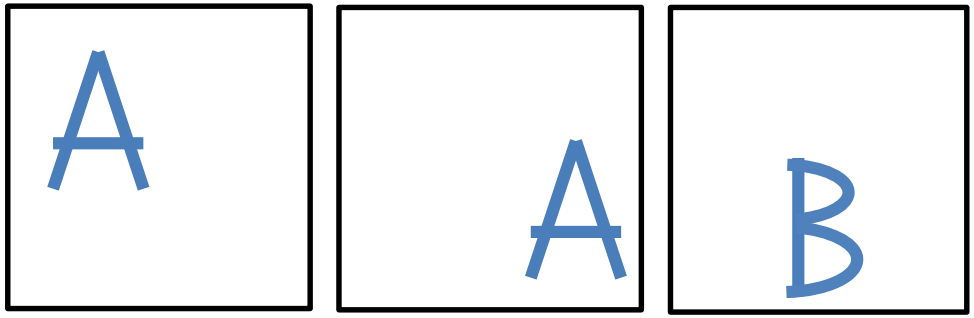
入力

復元画像

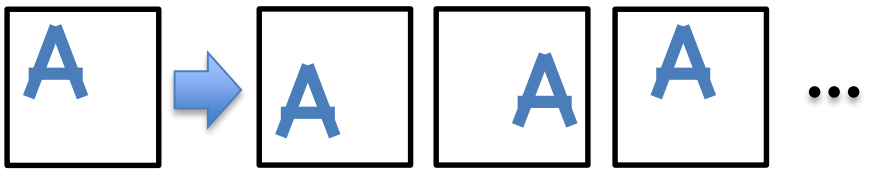


形状を復元後、推定した量だけシフトした画像

通常のauto-encoderの問題



同じ形状の画像は同一
descriptorにできないか？

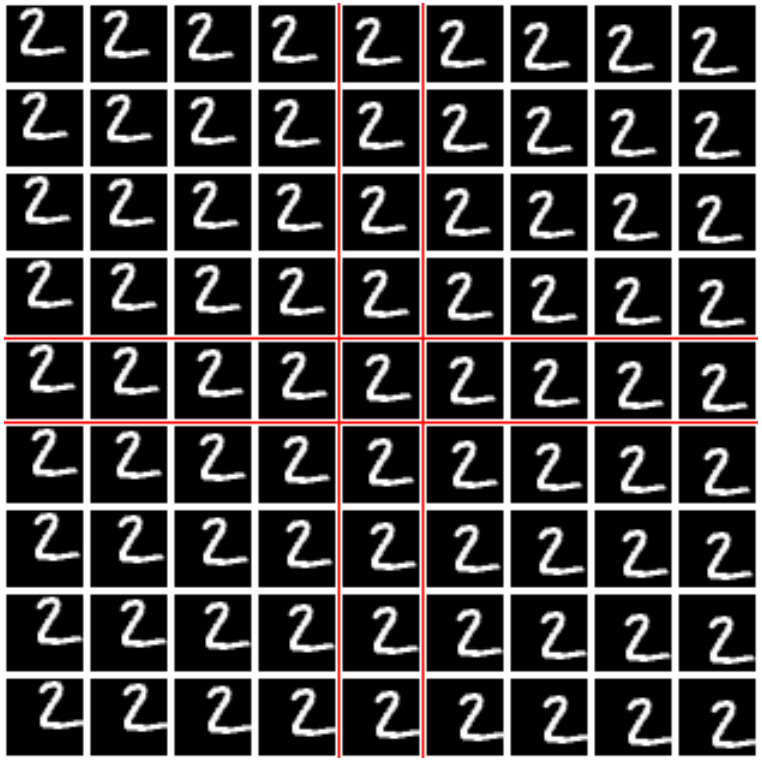


シフトした画像

形状は平行移動変換
（シフト）によって変化しない

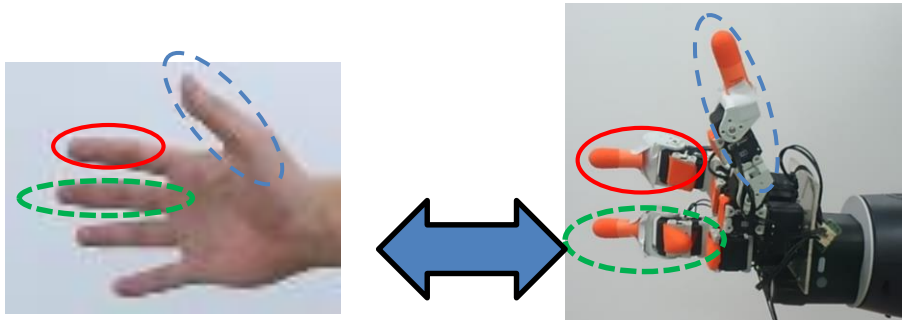
auto-encoderがシフトに関して不変で
あれば、形状そのものを表す
descriptorが得られる

形状復元とシフト量推定



入力

形状を復元後、推定した
量だけシフトした画像



Using shift invariant auto-encoder

Using PCA

RGB image

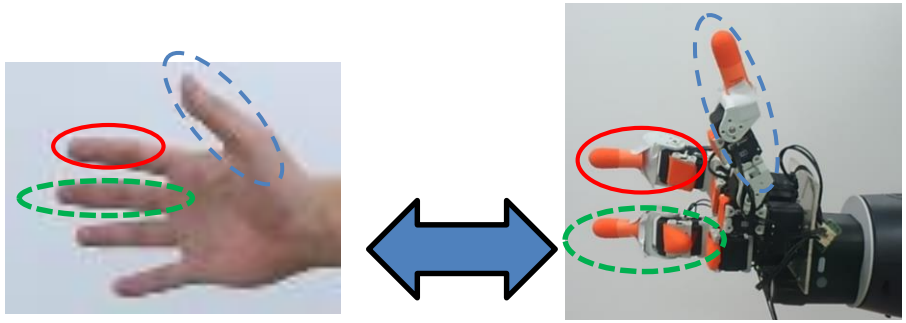
Depth image (input)

Restored depth image

Imitating robot hand

Restored depth image

Imitating robot hand



RITSUMEIKAN

Beyond Borders

変換不変auto-encoderの評価関数

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

不変性項 $C_{inv}(E,D)$

$$\sum_{I \in S} \sum_{i} \left\| D\left(E(I)\right) - D\left(E\left(T_{\theta_i}(I)\right)\right) \right\|_{L2}^2$$

入力Iを変換T_{θ_i}で変換してもencode/decode結果が変わらない制約

再生性項 $C_{res}(E,D)$

$$\sum_{I \in S} \min_{i} \left\| D\left(E(I)\right) - T_{\theta_i}(I) \right\|_{L2}^2$$

encode/decode結果は入力Iをいずれかの変換T_{θ_i}で変換したものに近いという制約

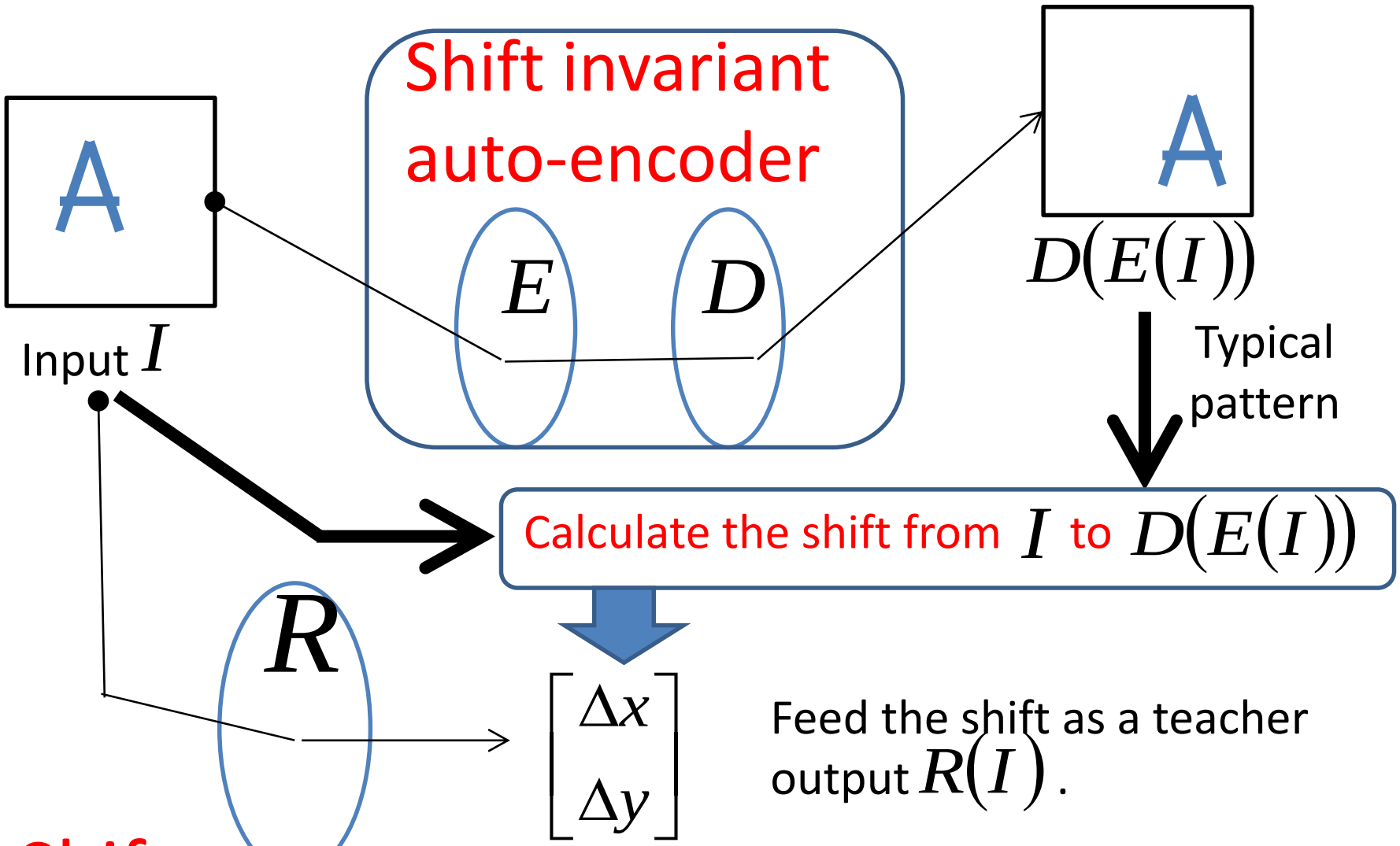
Sparseness項

$$C_{spa}(E)$$

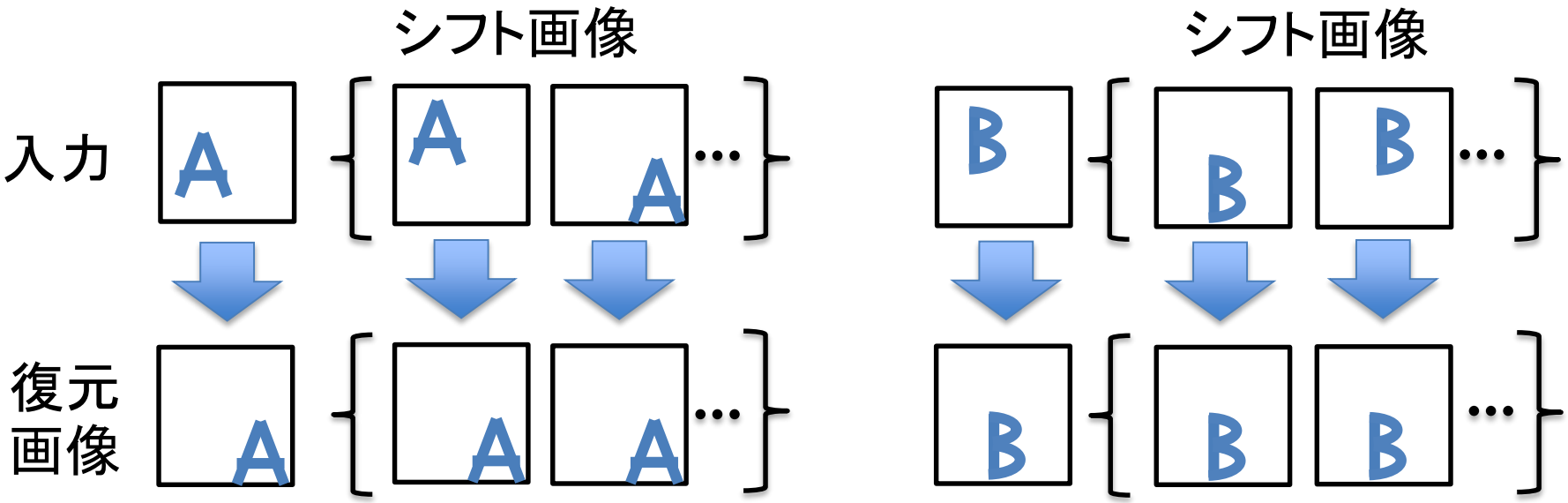
$$\sum_{I \in S} \frac{\|E(I)\|_{L1}^2}{\|E(I)\|_{L2}^2}$$

descriptor $E(I)$のエネルギーが少数の次元に集中しているべきという制約

シフト量推定器



再現性



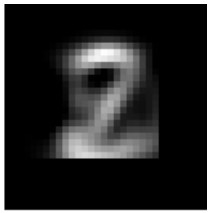
復元画像には形状が再現されるべき

提案法の概要

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）

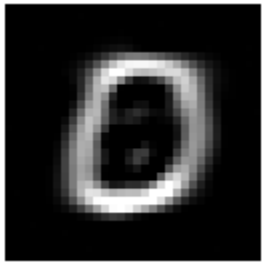
形状は不変



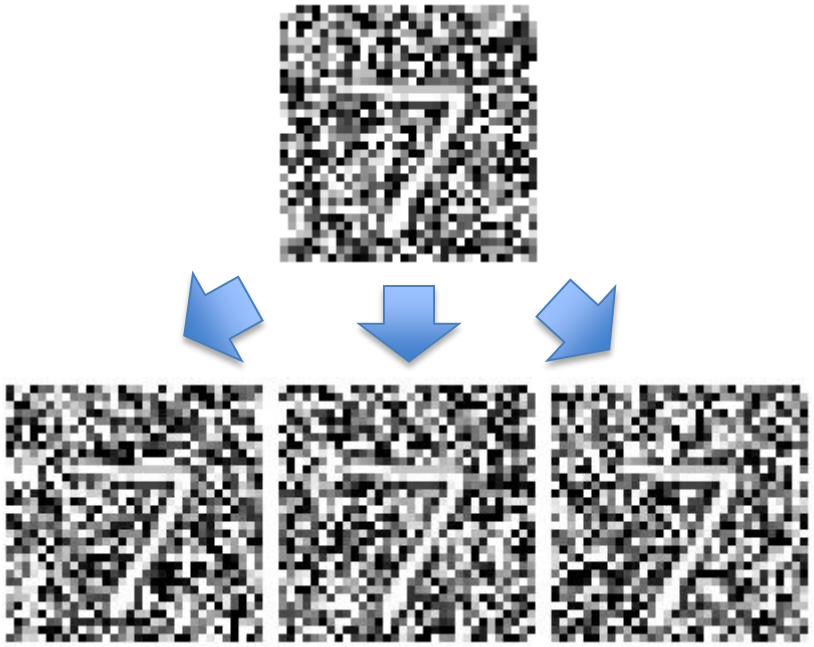
背景変更

前景は不変

- 教師ラベルや不変量の設計は不要
- 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



背景分離への適用



背景を変更する変換に関して不変なauto-encoderであれば前景領域を表すdescriptorが得られるはず

抽出された前景領域が見やすいよう、descriptorをdecodeすると背景領域の値が0となるよう制約する

変換不変auto-encoder

$$C(E,D) = \sum_{I \in S} \lambda_{\text{inv}} \sum_i \left\| D\big(E(I)\big) - D\left(E\left(T_{\theta_i}(I)\right)\right) \right\|_{L2}^2 + \lambda_{\text{res}} \min_i \left\| D\big(E(I)\big) - T_{\theta_i}(I) \right\|_{L2}^2$$

提案目的関数

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

不変性項 $C_{inv}(E,D)$

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

入力Iをいずれの変換T_θで変換しても
encode/decode結果が変わらない制約

再生性項 $C_{res}(E,D)$

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

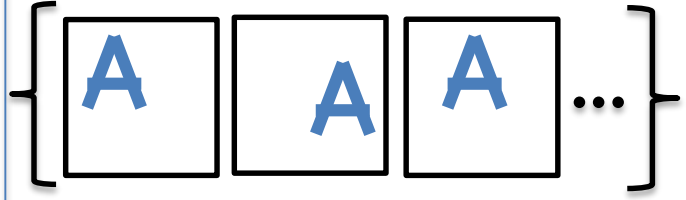
encode/decode結果は入力Iをいずれかの
変換T_θで変換したものに近いという制約

シフトの場合の例

入力 I

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

=



変換によって得られる
バリエーション

RITSUMEIKAN

Beyond Borders

特色あふれる
グローバル研究大学を目指して

立命館大学　広報課
立命 館太郎

2015.07.30

自己紹介

立命館大学
広報課

立命 館太郎

文字サイズ基本指標 見出し 42pt

文字サイズ基本指標 見出し 36pt

文字サイズ基本指標 見出し＆本文 30pt

文字サイズ基本指標 見出し＆本文 24pt

文字サイズ基本指標 本文 18pt

RITSUMEIKAN

Beyond Borders