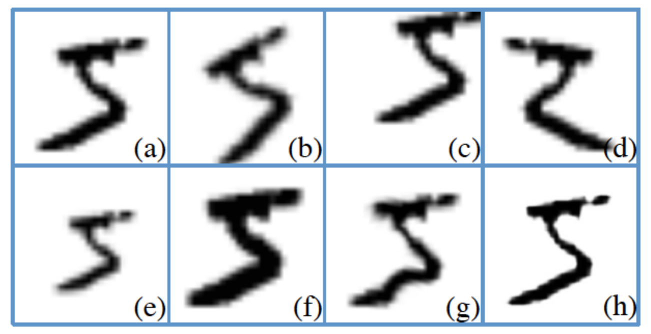
TI-pooling: transformation-invariant pooling for feature learning in CNNs

Dmitry Laptev*, Nikolay Savinov*, Joachim M. Buhmann, Marc Pollefeys
Department of Computer Science, ETH Zurich, Switzerland

1. Transformation invariance

Transformation invariances everywhere:

1. Natural images: illumination, camera view-point, projections.
2. Domain-specific, such as medical imaging: rotation, shift-invariances, scale, non-linear stretching, microscopy artefacts.
3. Computer-vision algorithms need to be robust to these variations, if the final result does not depend on them.



Human can easily recognize images under many different transformations: rotations, shifts, mirroring, scale, morphological operations, non-linear distortions, color change.

Related approaches:

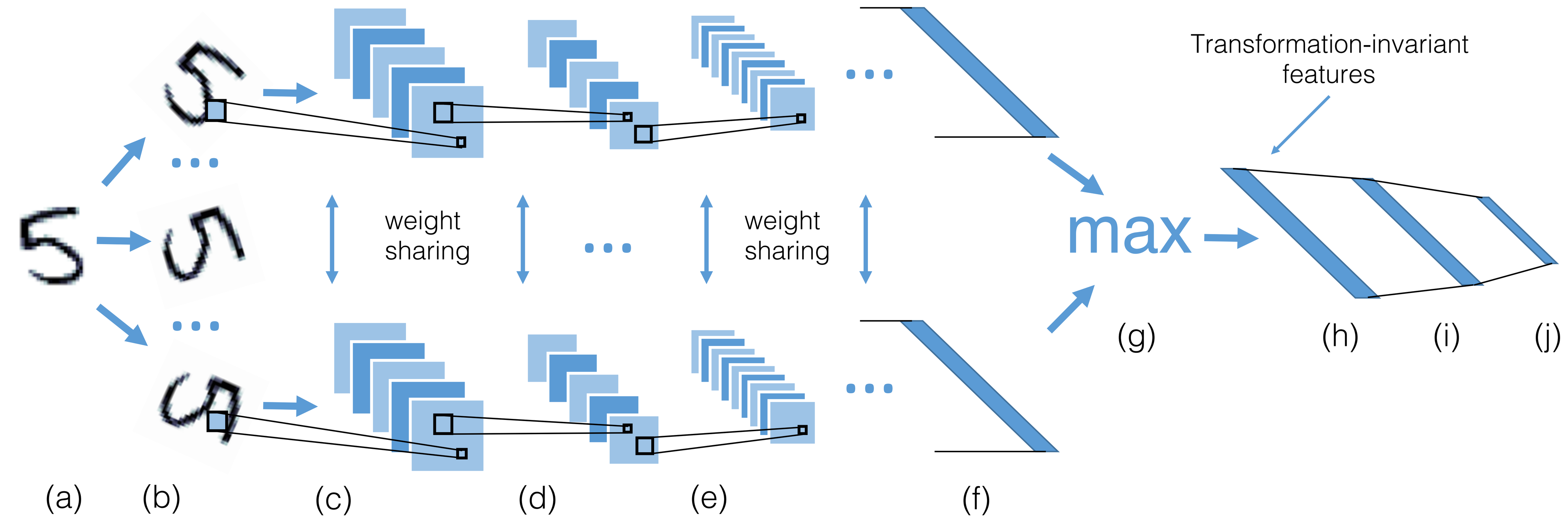
1. Predefined transformation-invariant features (SIFT, RIFT):
   - Allows for simple transformations, but not for arbitrarily-defined ones.
   - Can be used with very simple algorithms, but not with complex ones, such as deep learning.
2. Learning transformation-invariant features (TICJ):
   - Works with arbitrary transformations, but only with simple algorithms (decision trees / jungles).
3. Spatial Transformer Networks:
   - Good for learning transformations, not incorporating the known ones.
   - Introduces additional layer of complexity.
4. Multiple instance learning (multi-column networks):
   - The algorithm as a whole is transformation-invariant, but individual features are not.
5. Augmentation:
   - State of the art.
   - Relying on the flexibility of the network to learn a solution for every transformation.
   - Requires more flexible models.

TI-pooling:

1. Based on the combination of ideas from learning transformation-invariant features (2) and from multiple instance learning (4).
2. Guaranteed to learn transformation-invariant features for any arbitrary set of expert-defined transformations.
3. Allows to simplify the complexity and topology of the network, converges faster and more robust.

* The authors assert equal contribution and joint first authorship.

2. Proposed topology



- First, input image x (a) is transformed according to the considered set of transformations Φ to obtain a set of new image instances $\varphi(x)$, φ being from Φ (b).
- For every transformed image, a parallel instance of partial siamese network is initialized. Instances are passed through convolutional (c, e) and subsampling layers (d), until the vector of scalars is not achieved (e). It is composed of image features $f_k(\varphi(x))$ learned by the network.

$$g_k(x) = \max_{\phi \in \Phi} f_k(\phi(x))$$

- TI-pooling (g) is applied on the feature vectors as an element-wise maximum to obtain a vector of transformation-invariant features $g_k(x)$ (h). This vector then serves as an input to a fully-connected layer (i), possibly with dropout, and further propagates to the network output (j).
- Key components:
  - Siamese networks share parameters and therefore preserve the number of them.
  - Max allows for efficient gradient computations, therefore back-propagation works.

3. Experiments

Mnist-rot-12k

Method	Error, %
ScatNet-2 [3]	7.48
PCANet-2 [5]	7.37
TIRBM [25]	4.2
TI-POOLING (ours)	**2.2**

Half-rotated MNIST

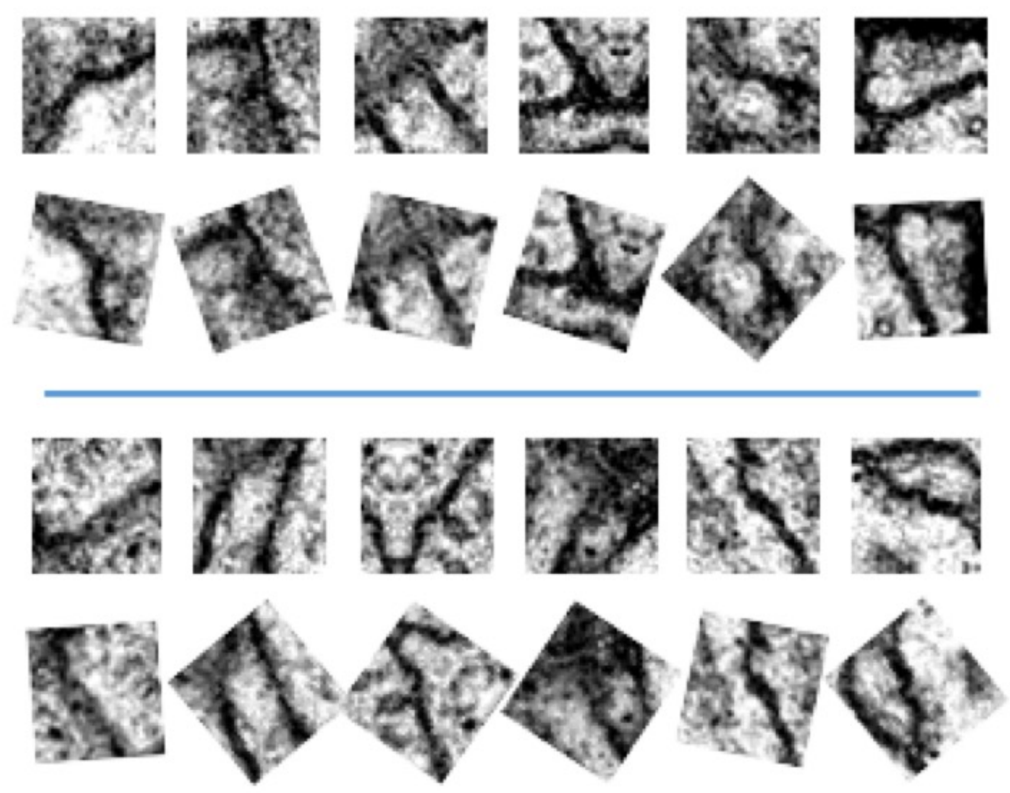
Method	Error, %
FCN	2.1
CNN	1.2
STN (general)	0.8
STN (affine)	0.7
TI-POOLING (ours)	0.8

Neuronal segmentation

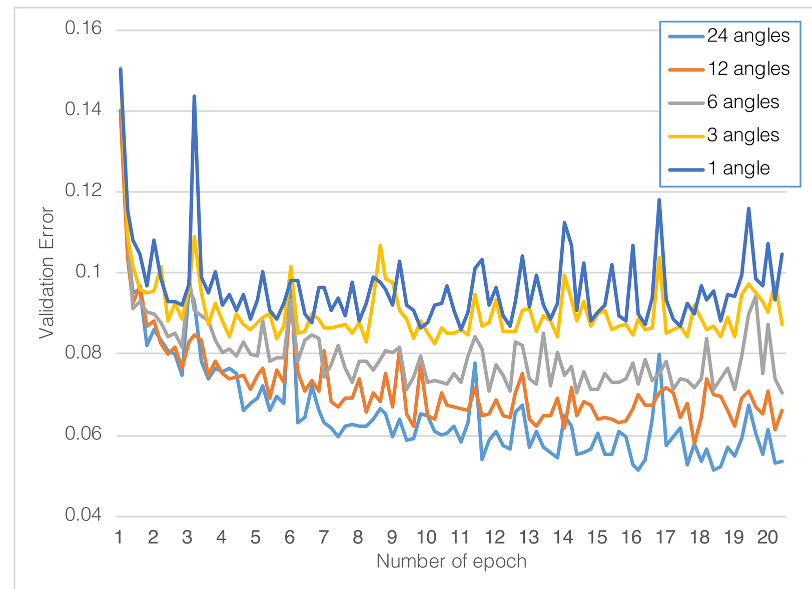
Method	Error, %
MIL over CNN [28]	8.9
CNN with augmentation [12]	8.1
TI-POOLING - dropout	**7.4**
TI-POOLING + dropout	**7.0**

See the paper for the detailed analysis and interpretation of the above results.

4. Properties



- Input patches from neuronal segmentation dataset (rows 1 and 3) together with the patches rotated by the maximum angle for some features $g_k(x)$.
- In most cases the membranes are oriented in approximately the same direction. The algorithm considers this orientation to be **canonical**.



- The more angles we sample for a set Φ – the better results are achieved (time=accuracy trade-off).
- Fewer canonical positions needs to be handled by the learning algorithm, unlike augmentation.

Lemma 1. The feature of the image x defined above is transformation-invariant if the set Φ of all possible transformations forms a group (axioms of closure, associativity, invertibility and identity).

Codes soon available at https://github.com/dlaptev