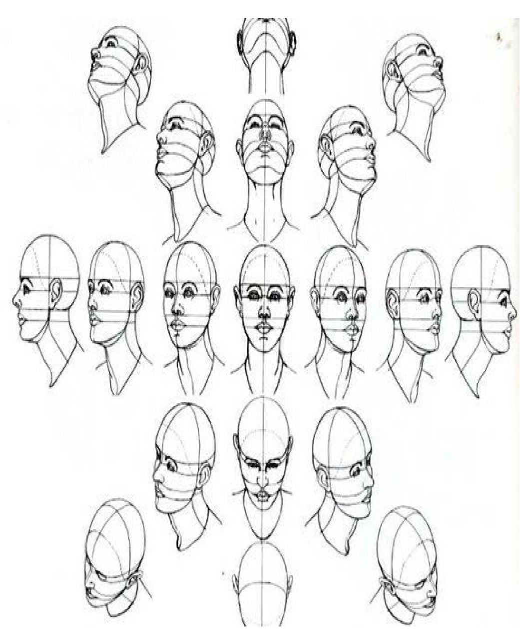
MINTIN: Maxout-based and Input-Normalized Transformation Invariant Neural Network

Jingyang Zhang, Kaige Jia, Pengshuai Yang, Fei Qiao, Qi Wei, Xinjun Liu, Huazhong Yang
Tsinghua University, Beijing
qiaofei@tsinghua.edu.cn

Backgrounds

➢Pictures taken in the real scenes have **variations** (rotation, shift, scaling , … …)



➢**Larger number of model parameters and longer training time** is required to make neural networks not sensitive to input variations by using data augmentation [1] or STN[2].

➢TI-Pooling[3] reports the state-of-art performance to solve it with smaller number of parameters. However, its computations is very large.

➢This work aims to improve TI-Pooling methods by **reducing computations, and also improve the performance.**

Key Features

➢Normalization

A **simple-to-implement** yet effective operation without heavy extra calculation, used to narrow down inputs' variations.

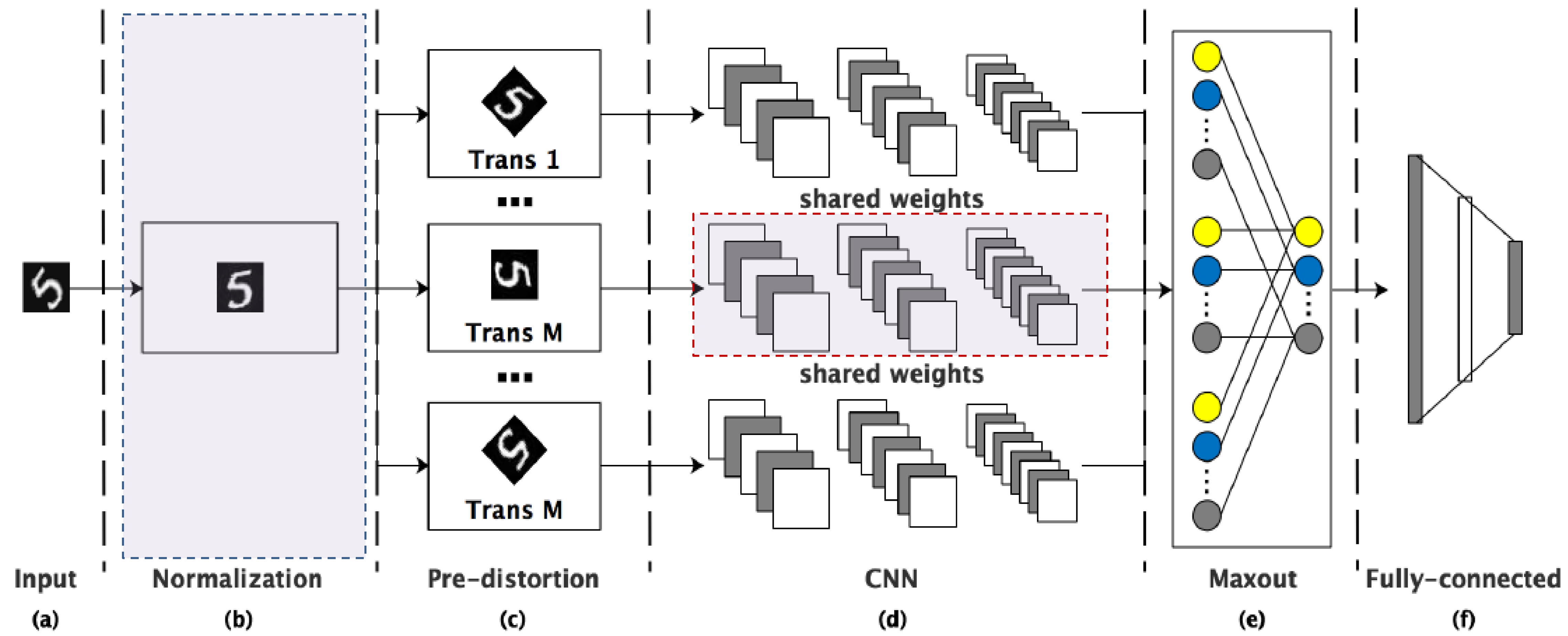
➢Cascade calibration

Divide the acquisition of invariance into two parts. **Normalization stage Focuses on each image itself but then Maxout stage focuses on the whole set of transformed versions.**

➢Dynamic and flexible frame

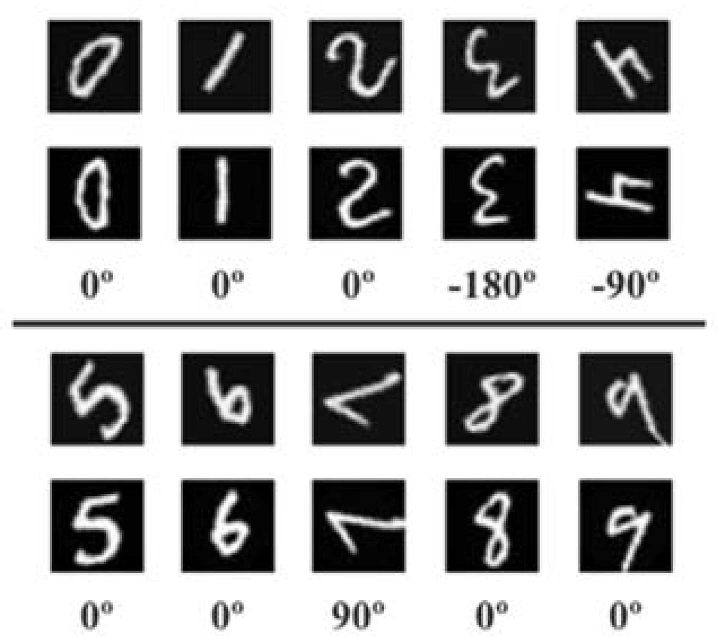
A **trade-off** between computations and accuracy. More channels means more computations and also higher accuracy.

Proposed Framework



➢ Normalization

Aims to **reduce the randomness of the transformation**, rather than to completely calibrate the input. Therefore, some simple algorithms, such as pixel-level statistical methods will be used.



The first and third rows show the original images, The second and fourth rows show the corresponding output of our Normalization module.

➢ Pre-distortion

Define **a set of transformations**, which reflects the possible present variations left in the dataset after normalization. Generating images then sent them into CNN for feature extraction.

➢ Multi-channel CNN and Maxout

Extracting the features and selecting the transformation invariant features among the transformation sets.

REFERENCES

[1] D. A. Van Dyk and X.-L. Meng, "The art of data augmentation," Journal of Computational and Graphical Statistics, vol. 10, no. 1, pp. 1–50, 2001.

[2] M. Jaderberg, K. Simonyan, A. Zisserman et al., "Spatial transformer networks," in Advances in Neural Information Processing Systems, 2015, pp. 2017–2025..

[3] D. Laptev, N. Savinov, et.al, "Ti-pooling: transformation-invariant pooling for feature learning in convolutional neural networks," in Proceedings of the IEEE CVPR, 2016, pp. 289–297

Experiment Results

Achieve **higher accuracy** than the state-of-art work with **less computations**.

➢ Rotation

- A statistical method based on **moment invariants** is applied to do Normalization

$$\mu_{pq} = \sum_{i=1}^{L}\sum_{j=1}^{H}(i-\bar{i})^{p}(j-\bar{j})^{q} f(i,j)$$

$$\tan 2\theta = \frac{2\mu_{11}}{\mu_{20}-\mu_{02}}$$

RESULTS ON MNIST-ROT-12K

Method	Channels	Error,%	Ops,$\times 10^7$	Relative Ops
TI-POOLING	4	2.47	5.14	0.167
	8	1.88	10.28	0.333
	24	1.61	30.84	1
MINTIN	4	1.76	5.14	0.167
	8	1.59	10.28	0.333
	24	1.57	30.84	1

RESULTS ON HALF-ROTATED MNIST

Method	Channels	Error,%	Ops,$\times 10^7$	Relative Ops
TI-POOLING	7	1.44	9.00	0.538
	13	1.46	16.71	1
MINTIN	7	1.32	9.00	0.538
	13	1.23	16.71	1

➢ Scale

- For the Normalization, we simply judge the boundary of the digits and then resizing the area of the digit to 20 x 20 pixels.

RESULTS ON SCALING MNIST

Method	Channels	Error,%	Ops,$\times 10^7$	Relative Ops
TI-POOLING	5	1.52	6.43	0.556
	9	1.32	11.57	1
MINTIN	5	1.01	6.43	0.556
	9	0.96	11.57	1