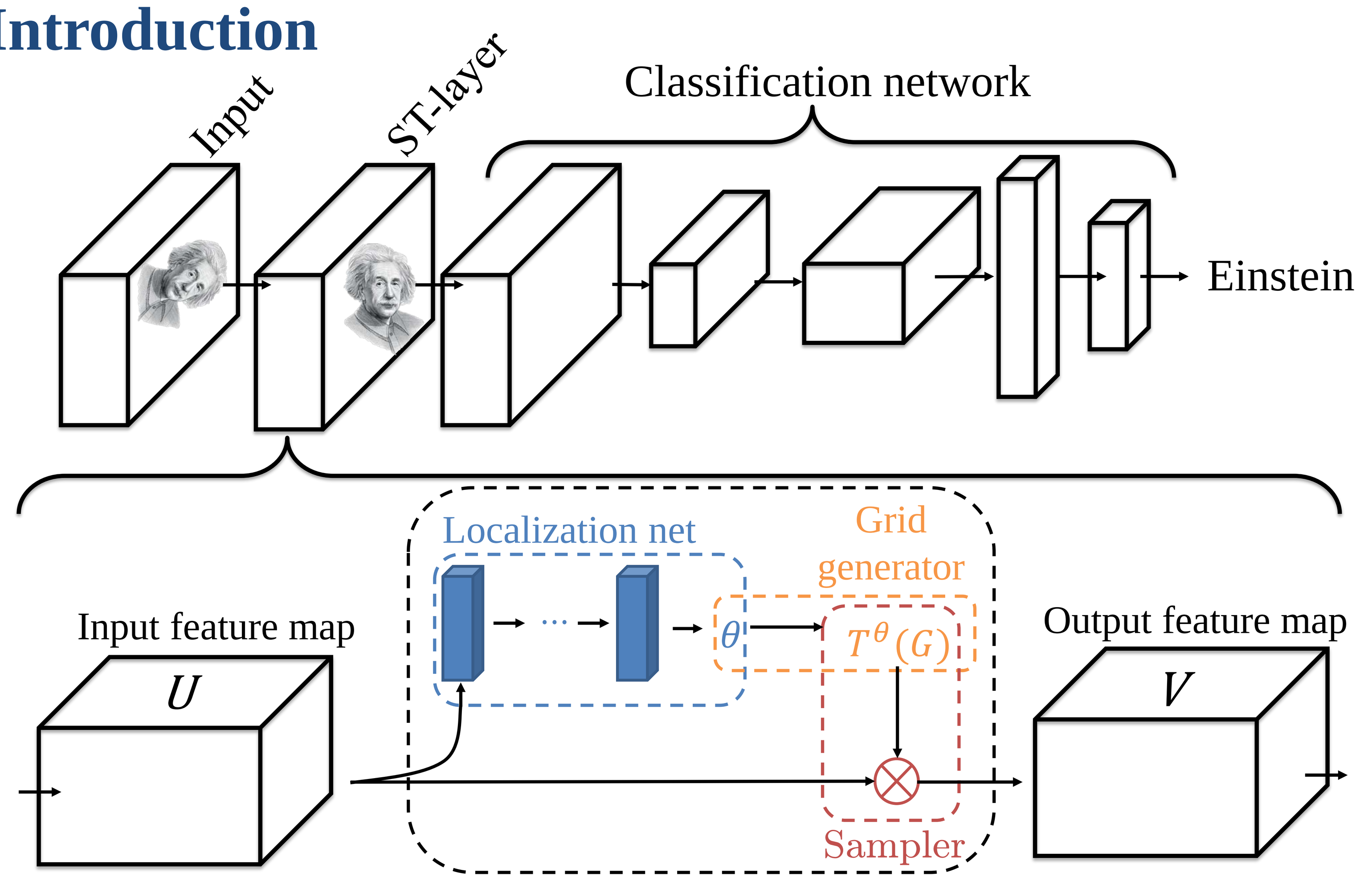
Deep Diffeomorphic Transformer Networks

Nicki Skafte Detlefsen[1] Oren Freifeld[2] Søren Hauberg[1]
nsde@dtu.dk orenfr@cs.bgu.ac.il sohau@dtu.dk
[1]Technical University of Denmark [2]Ben-Gurion University of the Negev

CVPR 2018 SALT LAKE CITY • JUNE 18-22

1. Introduction



Spatial Transformer layers[1] (ST-layer) allow neural networks to be *invariant* to large spatial transformation by learning input-dependent transformations.

Problem: Current implementations support transformations that are either too restrictive e.g. affine or homographic maps, and/or destructive maps, such as thin plate splines (TPS). These kind of transformation often leads to unstable optimization of networks.

2. Method - CPAB transformations

Our solution: Incorporate differentiable flexible diffeomorphisms (namely, invertible differentiable transformations with a differentiable inverse) in ST-networks

CPAB transformations[2-3]

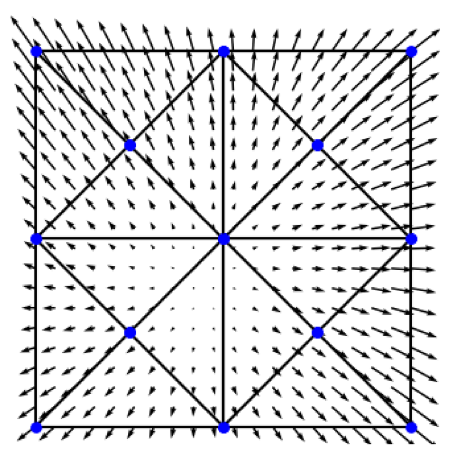
1. Divide image domain into Piecewise tessellation
2. Equip each cell with an Affine velocity field
3. Enforce constrains to make Continues velocity field v^θ
4. Based on v^θ, solve an integral equation (numerically)

$$\phi^\theta(x;1) = x + \int_0^1 v^\theta\left(\phi^\theta(x;\tau)\right)\, \mathrm{d}\tau$$

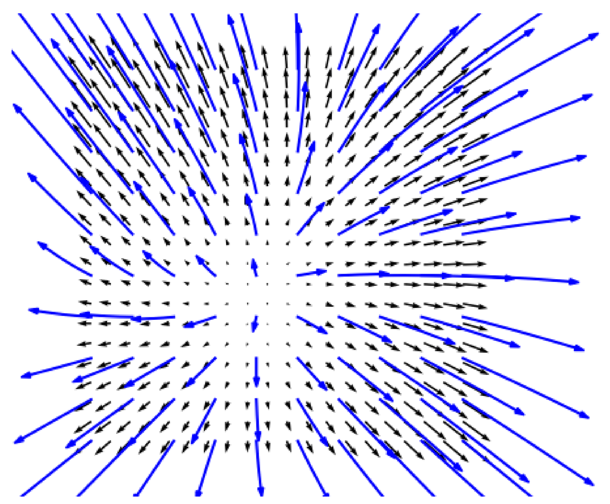
We have implemented an efficient algorithm for solving the integral in Tensorflow+CUDA.

Since the velocity field v^θ is Lipschitz continuous, our mapping $T^\theta(x) = \phi^\theta(x;1)$ is a diffeomorphism.

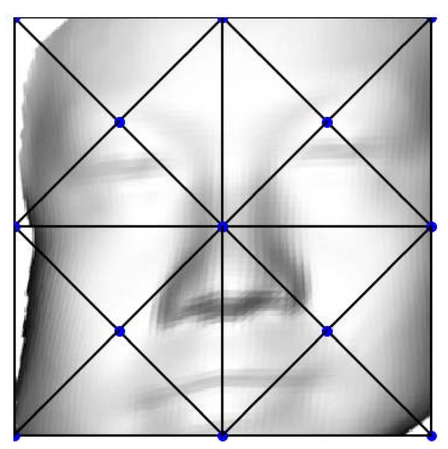
Continues piecewise affine v^θ



Trajectories



Mapping



3. Experiments

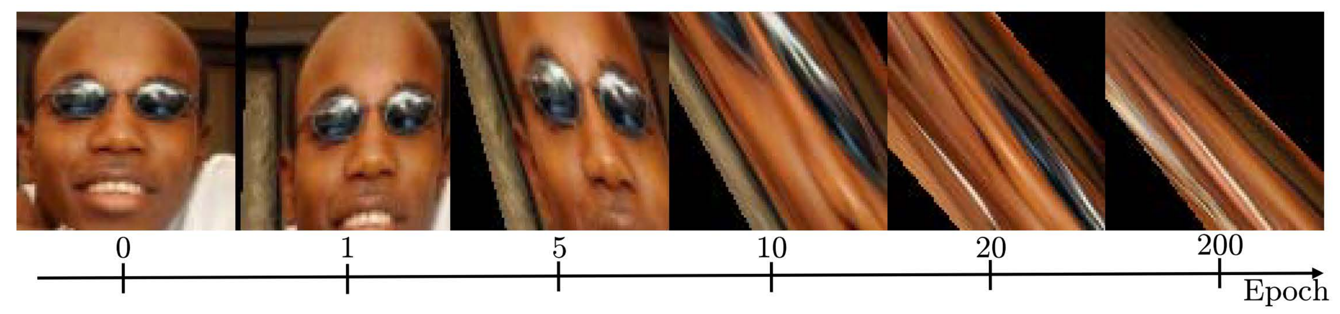
Selection of evaluated models:

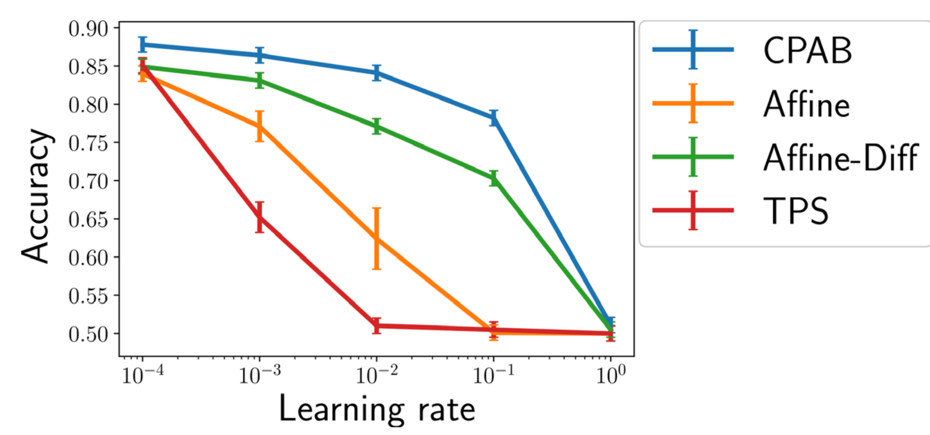
- Standard CNN
- CNN with an ST-Affine layer
- CNN with an ST-TPS layer
- CNN with an ST-CPAB layer
- CNN with an ST-Affine+CPAB layer

On the face datasets, the ST-CPAB yields a clear performance gain.
On the other datasets, there is no clear gain (but it also doesn't hurt)

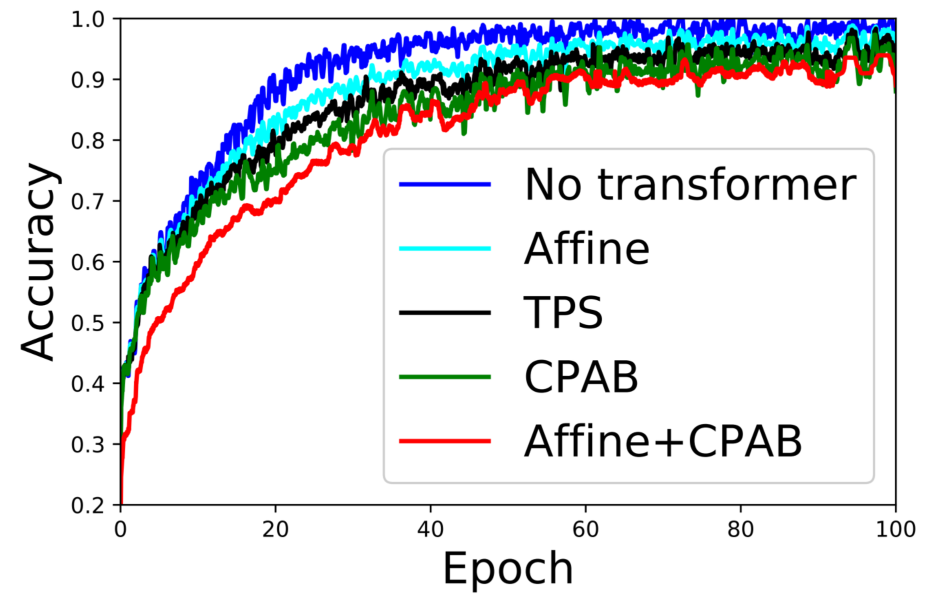
	Restricted LFW	CelebA	Unrestricted LFW
No transformer	0.788	0.712	0.893
ST-Affine	0.840	0.734	0.912
ST-TPS	0.851	0.742	0.921
ST-CPAB	0.878	0.751	0.936
ST-Affine+CPAB	0.893	0.756	0.954
State-of-the-art	**0.958**	**0.870**	**0.955**

Optimizing non-invertible ST-layers is prone to instability



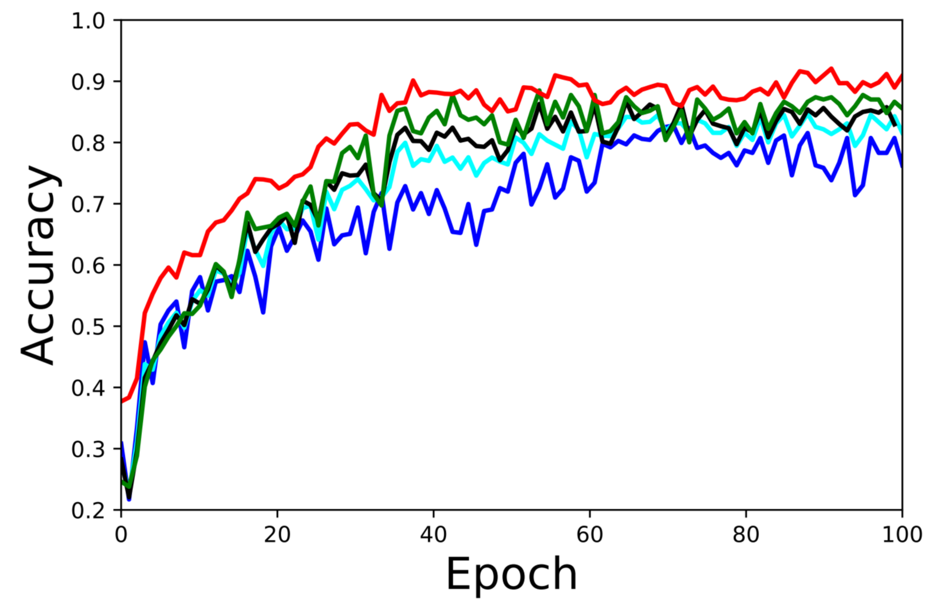


Invertible ST-layers are robust towards choice of a learning rate, compared with non-invertible ST-layers



←Training acc.
Validation acc.→

The proposed transformer model take more epochs to train, but eventually outperforms the others



Advantages of the proposed ST-layer

- Robustness
- Flexibility
- Low dimensional

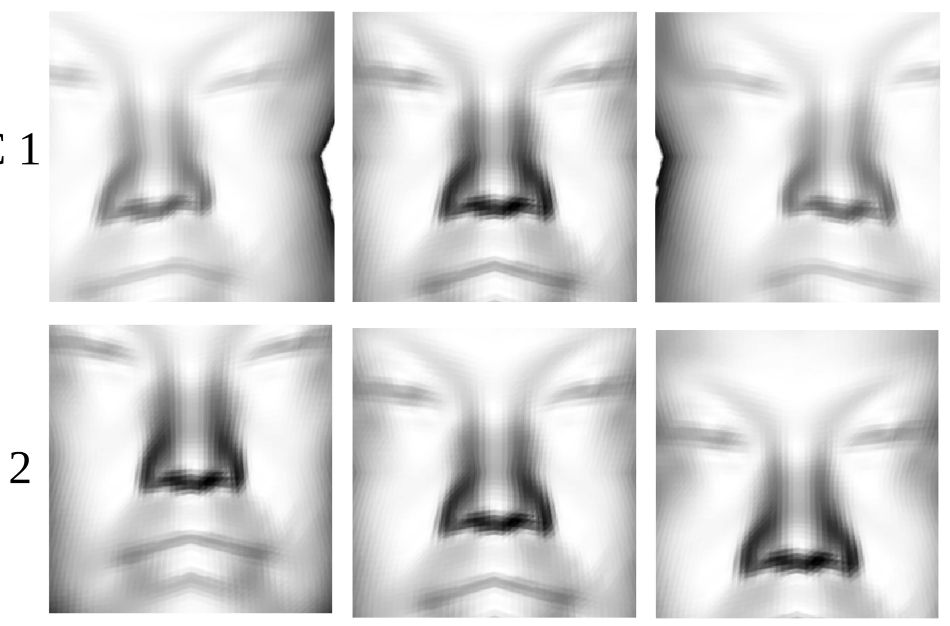
Disadvantage

- Computational times (only during learning)

Learned facial transformations mainly zoom in on the object in focus. The two leading principal components contribute to a vertical and horizontal translation of the images.

PC 1

PC 2



$-3\sigma_i$ 0 $+3\sigma_i$

Learning a "squarification" of the face, effectively remove the background

Transformed samples from different datasets

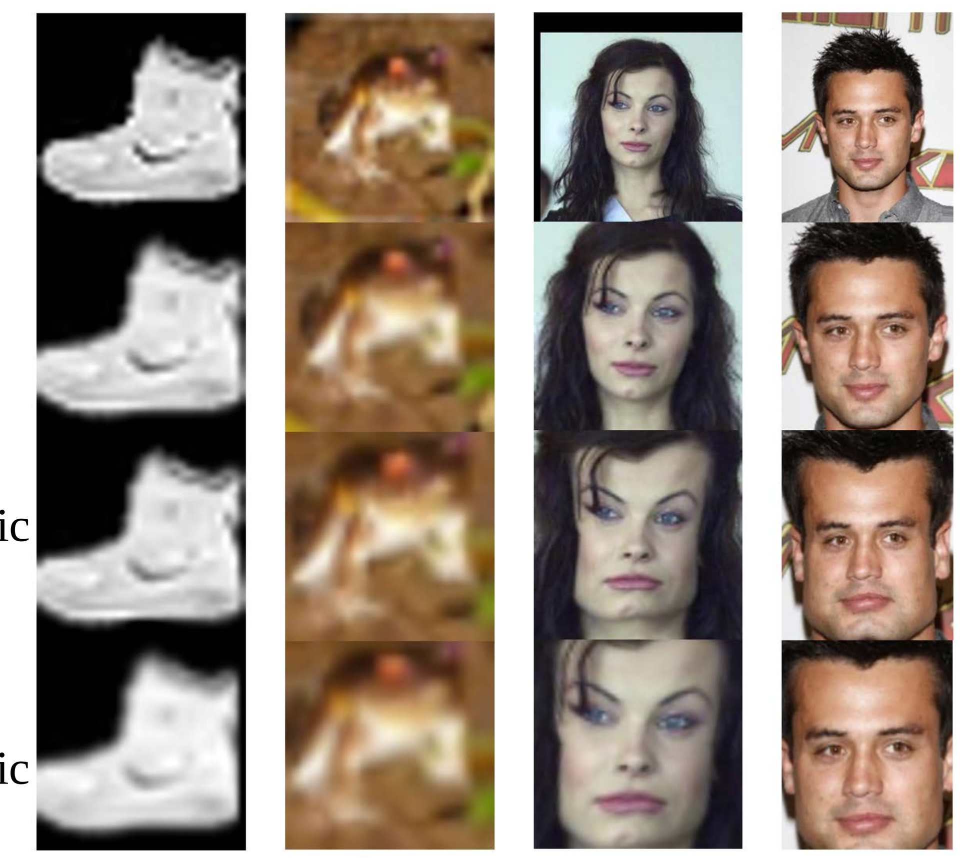
Fashion CIFAR LFW CelebA

Original

Affine

Diffeomorphic

Affine + Diffeomorphic



References

[1] M. Jadenberg, K. Simonyan, A. Zisserman and K. Kavukcuoglu, *Spatial transformer networks*, CoRR, 2015
[2] O. Freifeld, S. Hauberg, K. Batmanghelich, and J. W. Fisher III, *Highly-expressive spaces of well-behaved transformer: Keeping it simple.* In ICCV, 2015
[3] O. Freifeld, S. Hauberg, K. Batmanghelich, and J. W. Fisher III, *Transformations Based on Continuous Piecewise-Affine Velocity Fields.* In IEEE-TPAMI, 2017
[4] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, *Labeled faces in the wild: A database for studying face recognition in unconstrained environments.* Technical Report University of Massachusetts, Amherst, 2007

Get the code here
https://github.com/SkafteNicki/ddtn

4. Conclusion

The proposed diffeomorphic transformation have shown:

- Good optimization properties compared to standard ST transformations
- Intuitive "squarification" transformation, given insight into possible simple data augmentation scheme
- Performance gain facial benchmark tasks
- State-of-the-art on unrestricted LFW dataset[4]