Roto-Translation Equivariant Convolution Networks for Medical Image Analysis

Erik J Bekkers, Remco Duits

Dep. Math. & Computer Science CASA TU/e

Lie Analysis

Based on work by Bekkers[1], Lafarge[2], Veta[2], Eppenhof[2], Pluim[2], Duits[1] (MICCAI 2018)

_____2019/05/04_____Imaging & Machine Learning_____Institut Henri Poincaré, Paris, France_____

[1]Department of Mathematics and Computer Science, and

[2]Department of Biomedical Engineering,

Eindhoven University of Technology

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Presentation outline

Presentation outline

- Motivation 1: Geometric image analysis via orientation scores needs automation.
- Motivation 2: Machine learning needs group equivariance.

- Overview of related work.

- Theoretical background:
  - Neural networks (NNs)
  - Convolutional neural networks (CNNs)
  - Group convolutional neural networks (G-CNNs)

    General Theorem on Equivariant linear operators

    ⇩

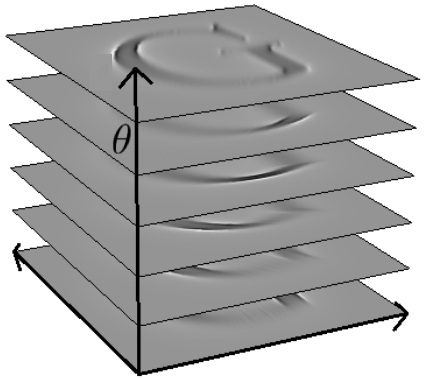
    Architecture G-CNNs.

- Results on 3 different medical imaging applications.
- Conclusion.

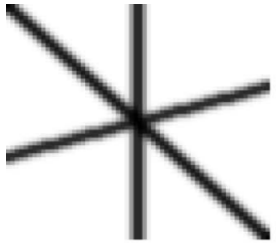
Equivariant image analysis
via orientation scores

Invertible Orientation scores

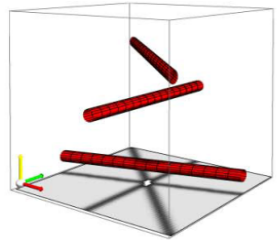
Image



Orientation Score (real part)

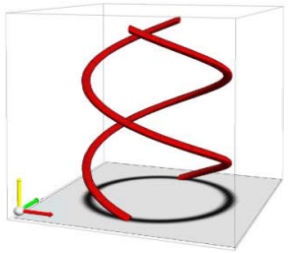


Image



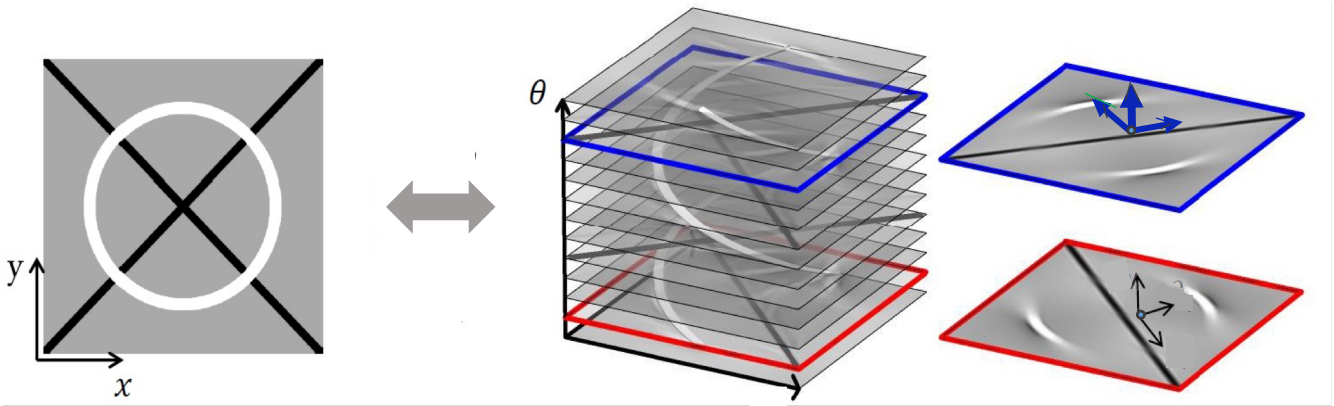
OS

Image



OS

Curved & Torqued Geometry of roto-translation group SE(2) visible in Score:

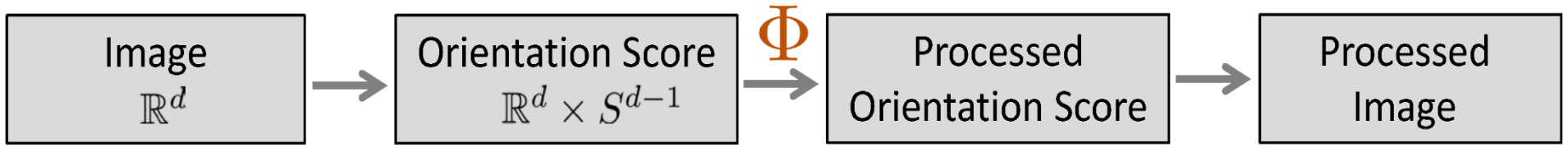


Moving frame → PDEs For tracking and regularization

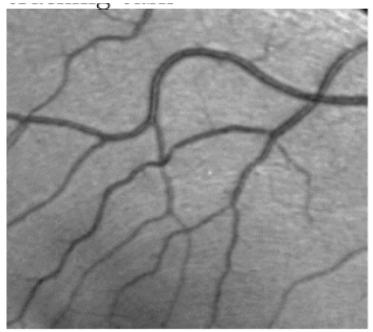
$$\mathcal{A}_1 = \cos\theta\,\partial_x + \sin\theta\,\partial_y$$
$$\mathcal{A}_2 = -\sin\theta\,\partial_x + \cos\theta\,\partial_y$$
$$\mathcal{A}_3 = \partial_\theta$$

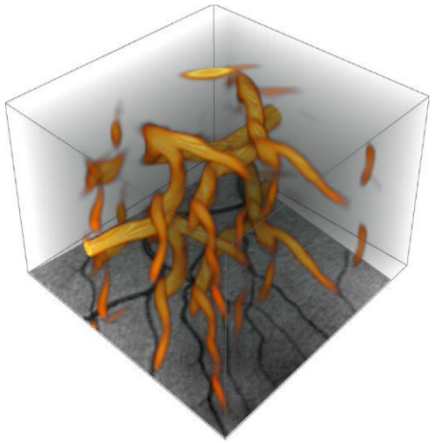
PhD Thesis	**R. Duits**	**2005**
PhD Thesis	**E.M. Franken**	**2009**
PhD Thesis	**E.J. Bekkers**	**2017**
PhD Thesis	**J.M. Portegies**	**2018**

SE(d) equivariant processing via orientation scores



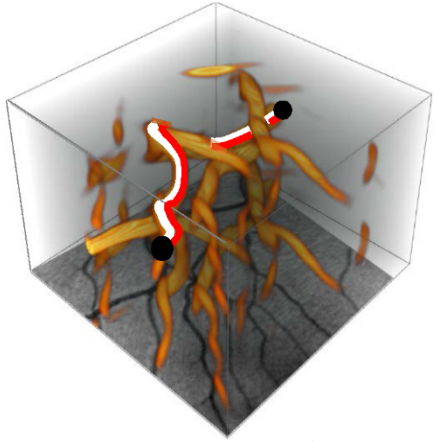
$d = 2$. Tracking Task.

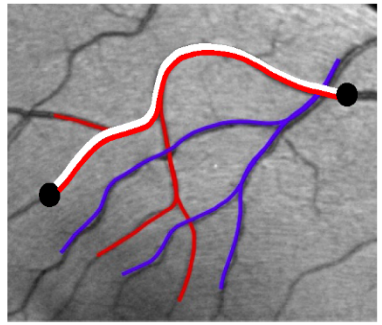




A.

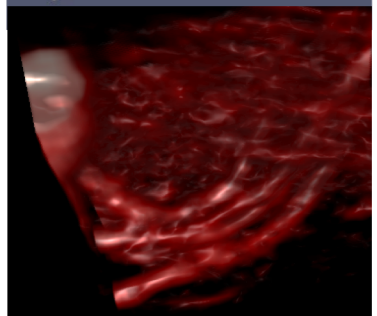
θ

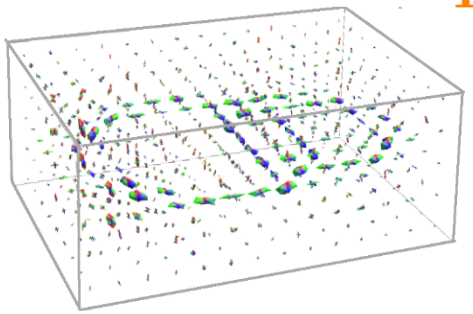




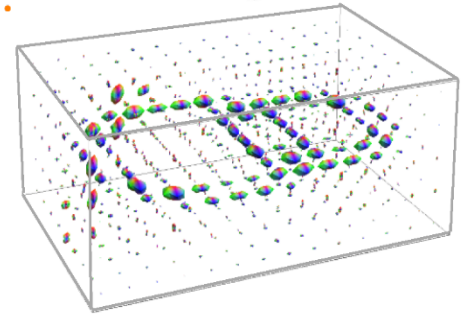
JMIV 2018
SIIMS 2016
JDCS 2016

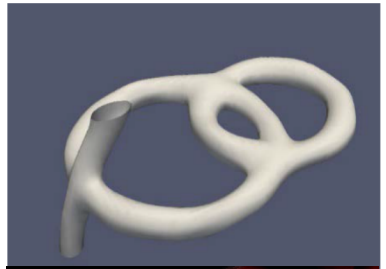
$d = 3$. Task:

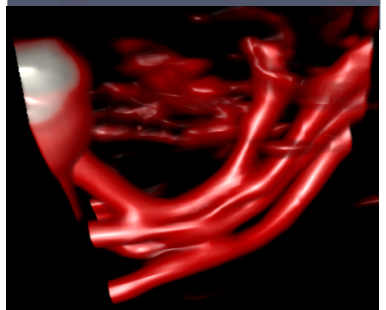




B.







JMIV 2018

A. Tracking via Optimal Geodesics

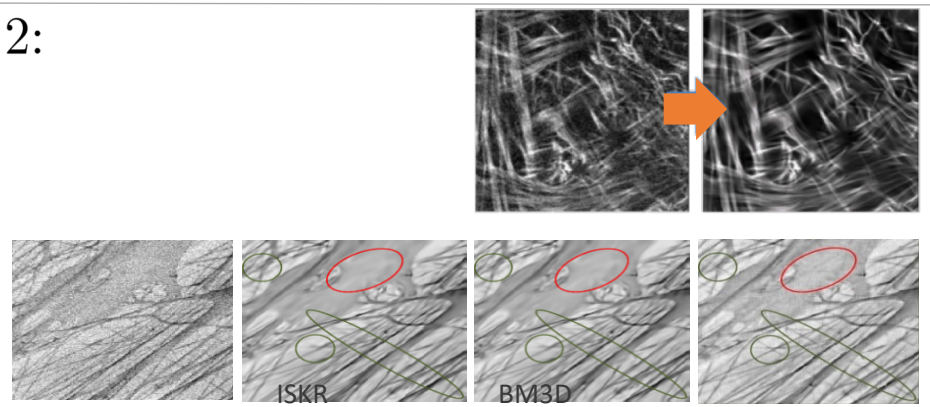
B. Enhancement via PDE-flows: Diffusion, MC or TV on $\mathbb{M} = \mathbb{R}^d \rtimes S^{d-1}$

Diffusion & Brownian Motions in Roto-Translation Group

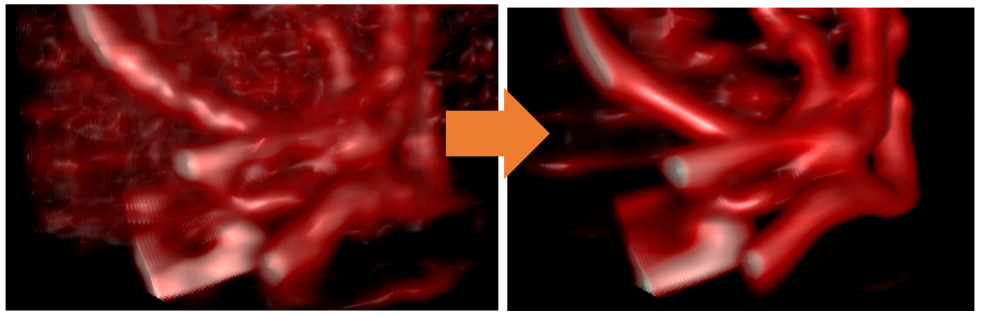
2nd Workshop IHP: "PDEs on the Homogeneous Space of Positions and Orientations." R. Duits

Numerics: Nonlinear Diffusions, 2009–2010

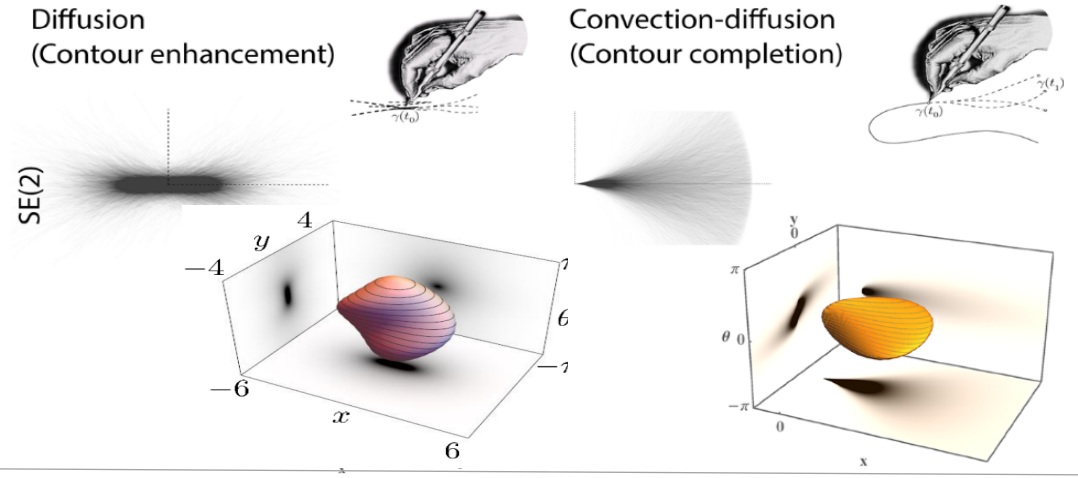
$d = 2$:



$d = 3$: 2018

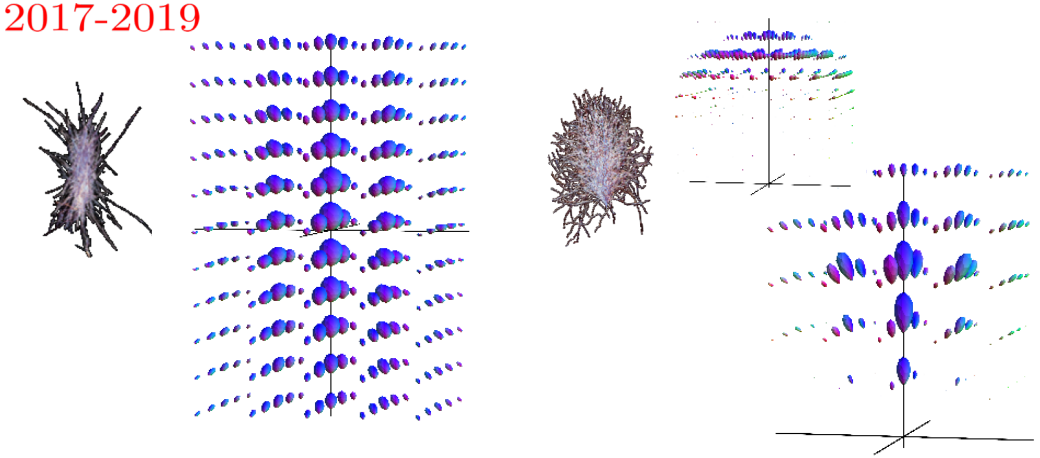


Exact: Linear (convection-)diffusion, 2005-2010,
SOLVED BY GROUP CONVOLUTION WITH PROBABILITY KERNELS



QAM-AMS
2008, 2010

2017-2019



DGA 2018

Entropy 2019

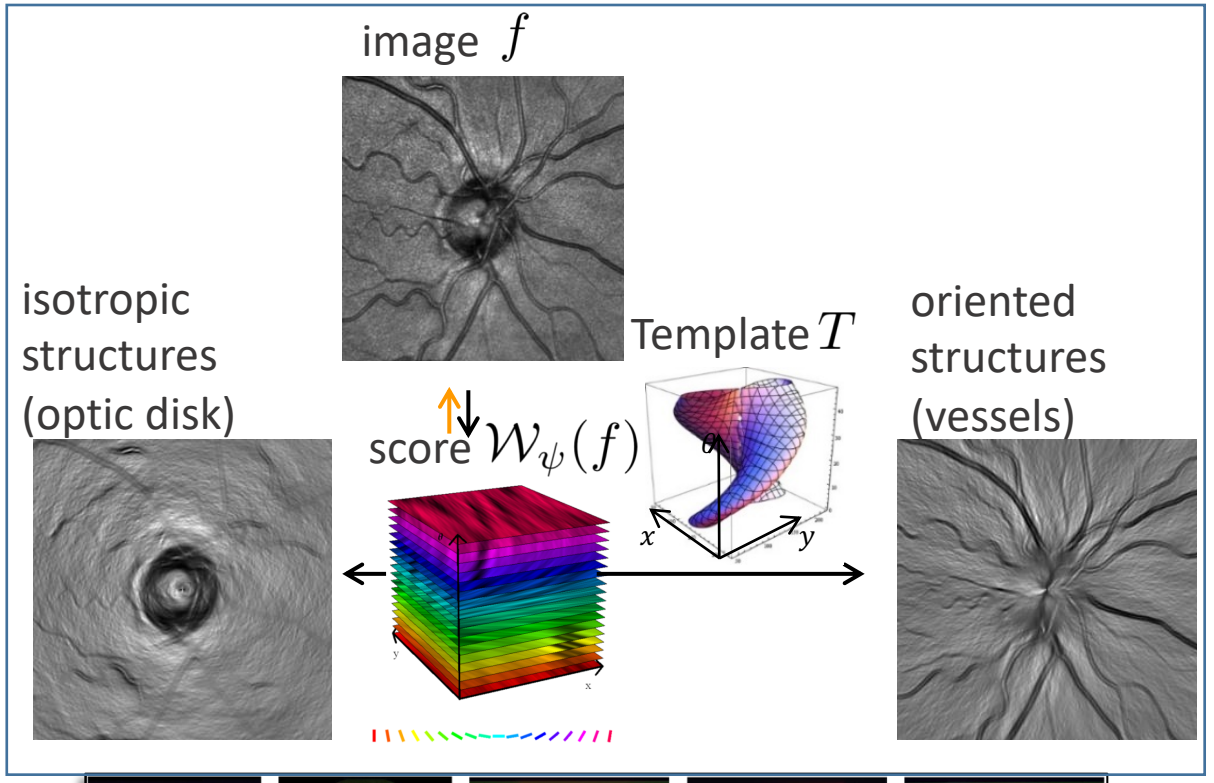
Next: Machine Learning to train such kernels and their offsets.
2014-onwards incl. IEEE-PAMI E.J. Bekkers & R. Duits et al.

Template matching via Group convolutions

ICIAR 2014 & PAMI 2018 Bekkers-Loog-tHRomeny-Duits

State-of-the-Art Results (99,8 % success rate) on 3 Detection Tasks. E.g. Optic Disc Detection via Template Matching using group convolutions in SE(2)

Special Case 2 layer Group CNN !



$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \max_{\theta} \left(\mathcal{L}_{g=(\mathbf{x},\theta)} T, \mathcal{W}_\psi f\right)$$

T minimizes $\mathcal{E}(T)$.

2 layer G-CNN: ICIAR 2014 & PAMI 2018 Bekkers-Loog-Duits

- Minimize Functional:

$$\mathcal{E}(T) = \sum_{i=1}^{N} | \, (T, U_{f_i}) - y_i \, |^2 + \int_{SE(2)} \lambda \, \|\nabla_{SR} T\|^2 + \mu \, |T|^2 \, \mathrm{d}g$$

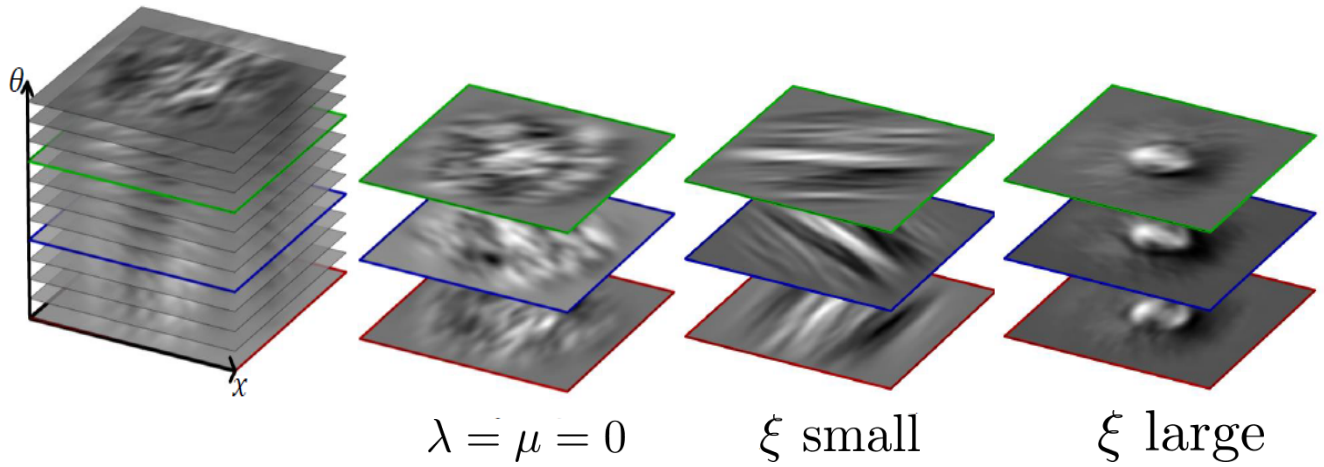
training patch; label

- With Sub-Riemannian Brownian prior

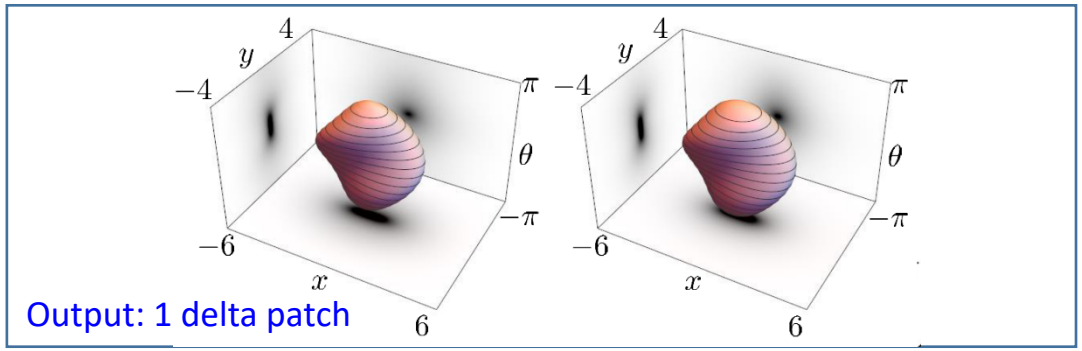
$$\int_{SE(2)} \|\nabla_{SR} T\|^2 \, \mathrm{d}g = \int_{SE(2)} \xi^2 \, |(\cos\theta \, \partial_x + \sin\theta \, \partial_y) T|^2 + \left| \frac{\partial T}{\partial \theta} \right|^2 \mathrm{d}g$$

Convex-problem.

Train parameters via
Generalized Cross-Validation:



- Expansion in B-splines



Accurate: Comparison to exact sol's Duits

Motivation 1

Extend from 2 Layer G-CNN for template matching in OS
to
Deep Learning in OS

Related work

Group equivariant networks

TU/e

Group convolution networks (domain extension)

LeCun et al 1990	$\mathbb{Z}^2$	translation networks
Mallat et al. 2013, 2015	SE(2)	Scattering transform & SVM
Bekkers et al. 2014-2018	SE(2)	via B-splines, 2 layer G-CNN
Cohen-Welling 2016	p4m	via 90° rotations + flips + theory!
Dieleman et al. 2016	p4m	via 90° rotations + flips
Weiler et al. 2017	SE(2)	via circular harmonics
Zhou et al. 2017	SE(2)	via bilinear interpolation
Bekkers et al. 2018	SE(2)	via bilinear interpolation
Hoogeboom et al. 2018	S(2,6)	hexagonal grids
Winkels-Cohen 2018	SE(3,N) + m	90° rotations + flips
Worrall-Brostow 2018	SE(3,N)	90° rotations
Cohen et al. 2018	SO(3)	via spherical harmonics

Steerable filter networks (co-domain extension)

Worrall et al. 2017	SE(2)	irrep
Marcos et al. 2017	SE(2)	vector field networks
Kondor 2018	SE(3)	irrep, N-body nets
Thomas et al. 2018	SE(3)	irrep, point clouds
Weiler et al. 2018	SE(3)	irrep
Esteves	SO(3)/SO(2)	irrep
Kondor-Trivedi 2018	SO(d)	irrep (on compact quotient sp.)

Continuous

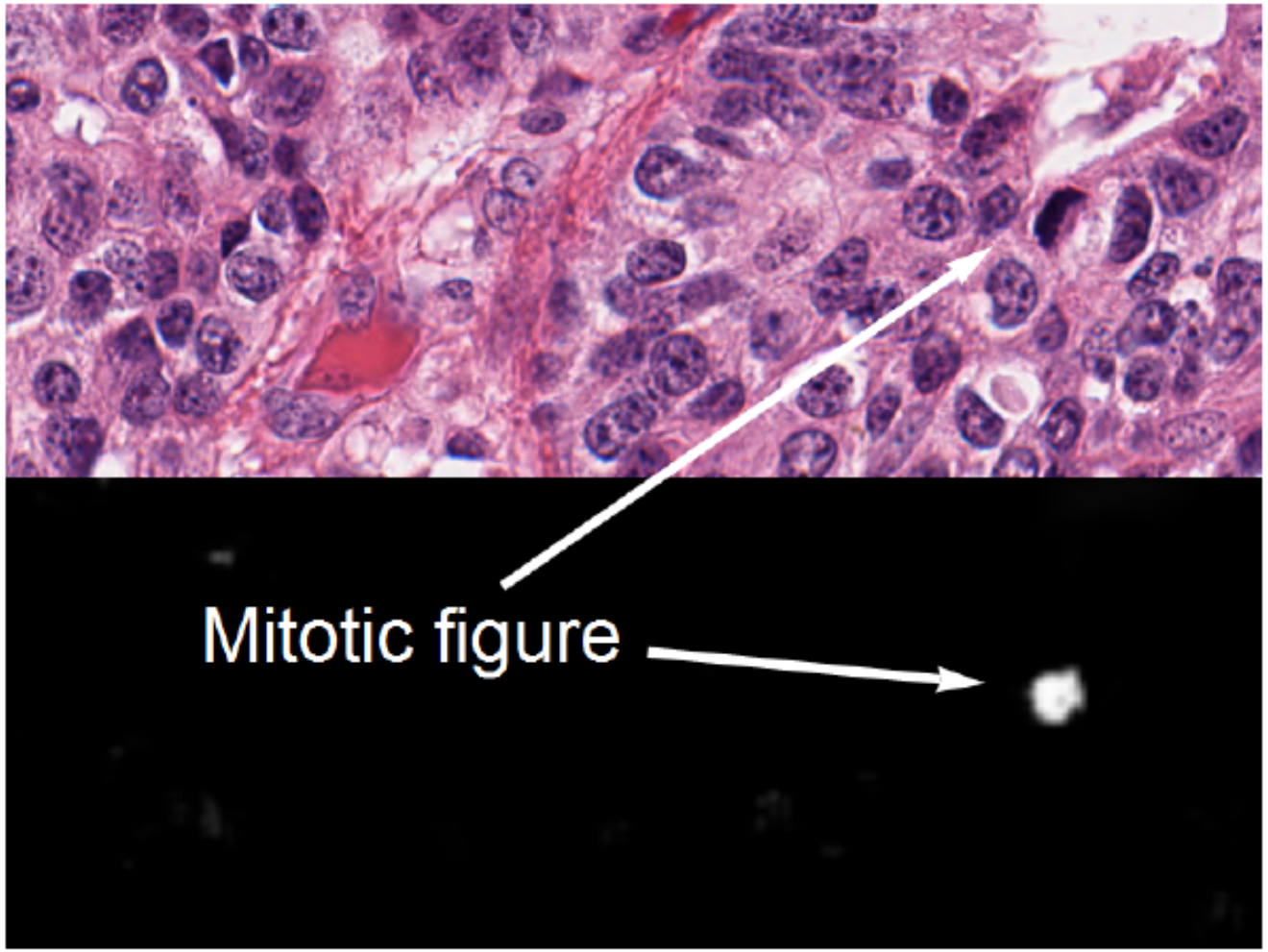
Discrete

Motivation 2

Roto-Translation Covariant Convolution Neural Networks for Medical Image Analysis

Many computer vision and MedIA problems require invariance/equivariance properties (e.g. w.r.t. rotation)

Rotation-invariant detection of pathological cells (mitotic figure) in histopathology



Transformation invariance

Classification via CNN:

CNN

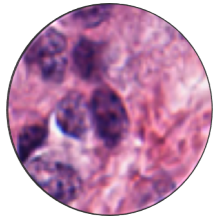
$$y = \Phi_{\underline{\omega}}(x)$$

Label (e.g. 1: healthy cell, 0: pathological cell)

Input image

Transformation:

Geometric transformation by $\mathcal{T}_g$ (e.g. rotation)



x

$\Phi_{\underline{\omega}}(x)$

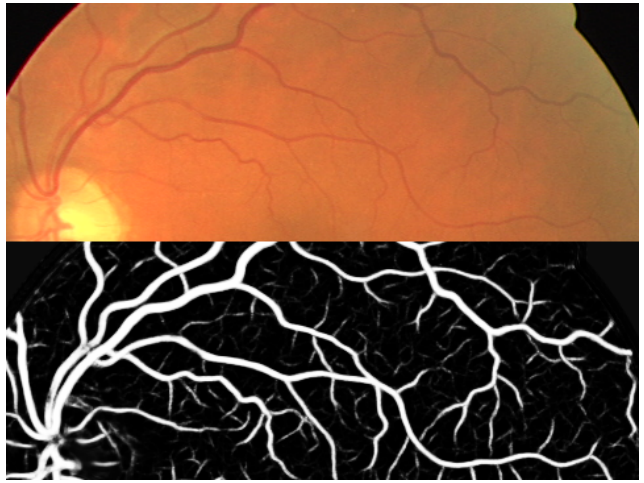
"healthy cell"

$\Phi_{\underline{\omega}}(\mathcal{T}_g x)$

$\mathcal{T}_g x$

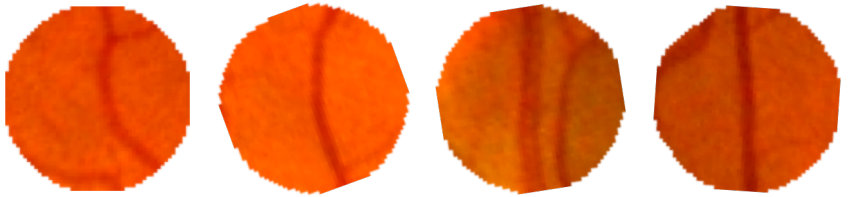
Transformation invariance by data augmentation

Example: vessel segmentation



Training set (*naïve example of only vertically aligned vessels*)

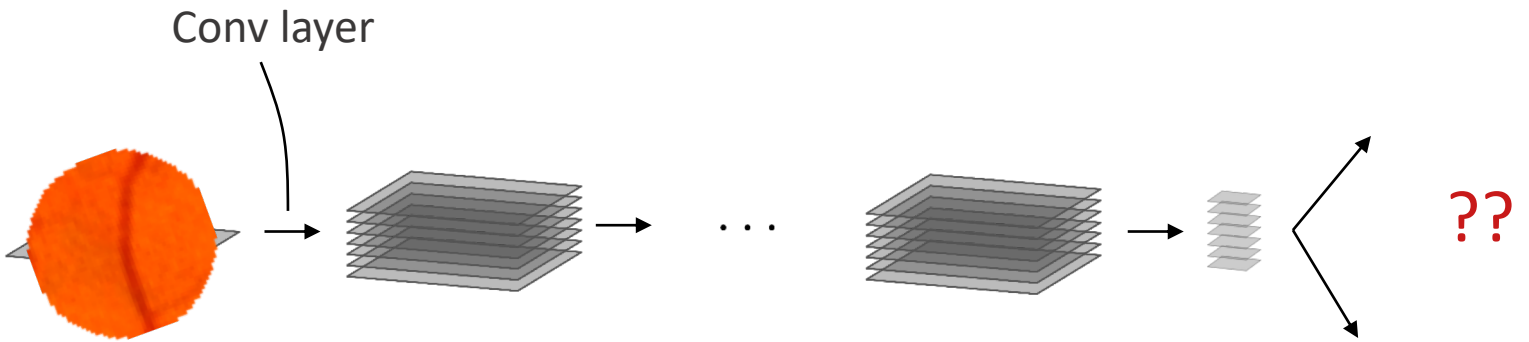
Blood vessels
(vertically aligned)



Background

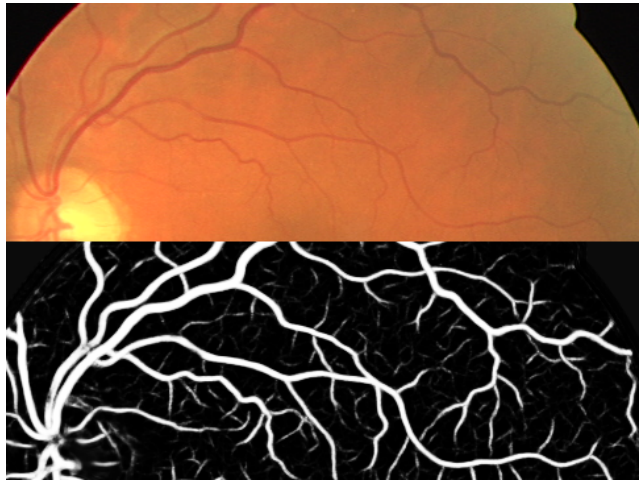
A simple network will do

$$x \xrightarrow{\Phi_{\omega}} y$$



Transformation invariance by data augmentation

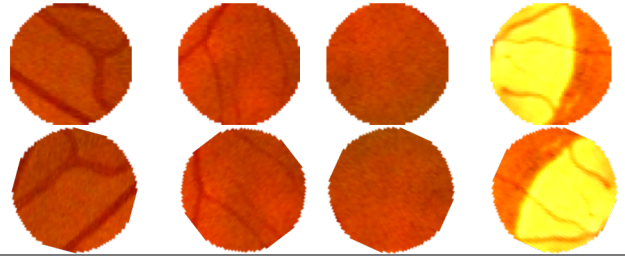
Example: vessel segmentation



Expand training set with rotated copies -> data augmentation

Blood vessels

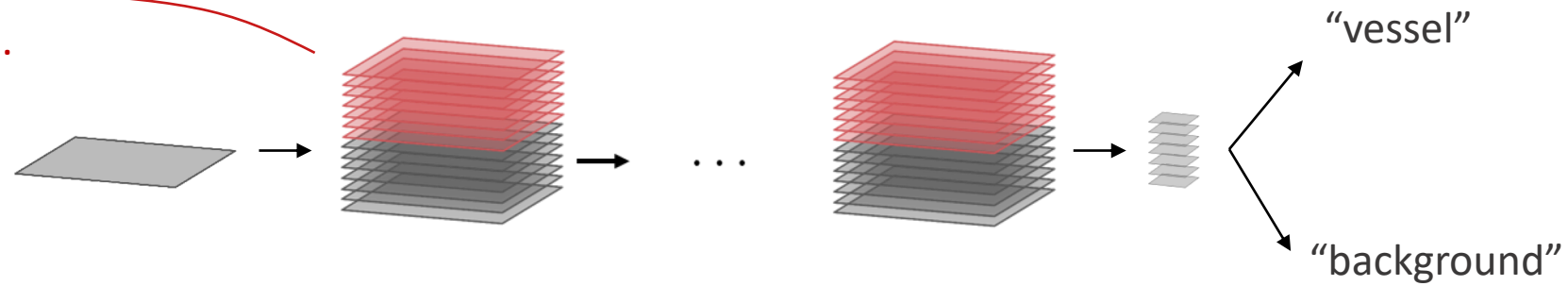
Background



A more complex network is required

$$x \xrightarrow{\Phi_{\underline{\omega}}} y$$

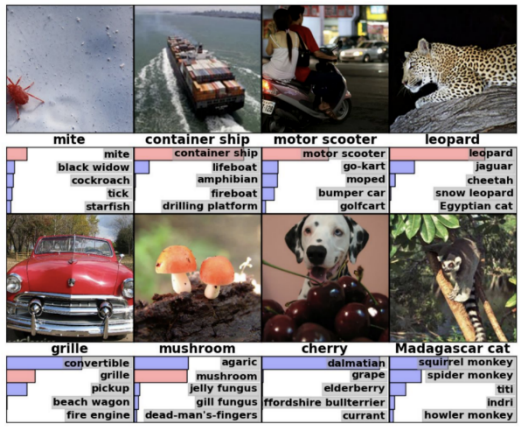
Now you learn rotated versions of the same feature..



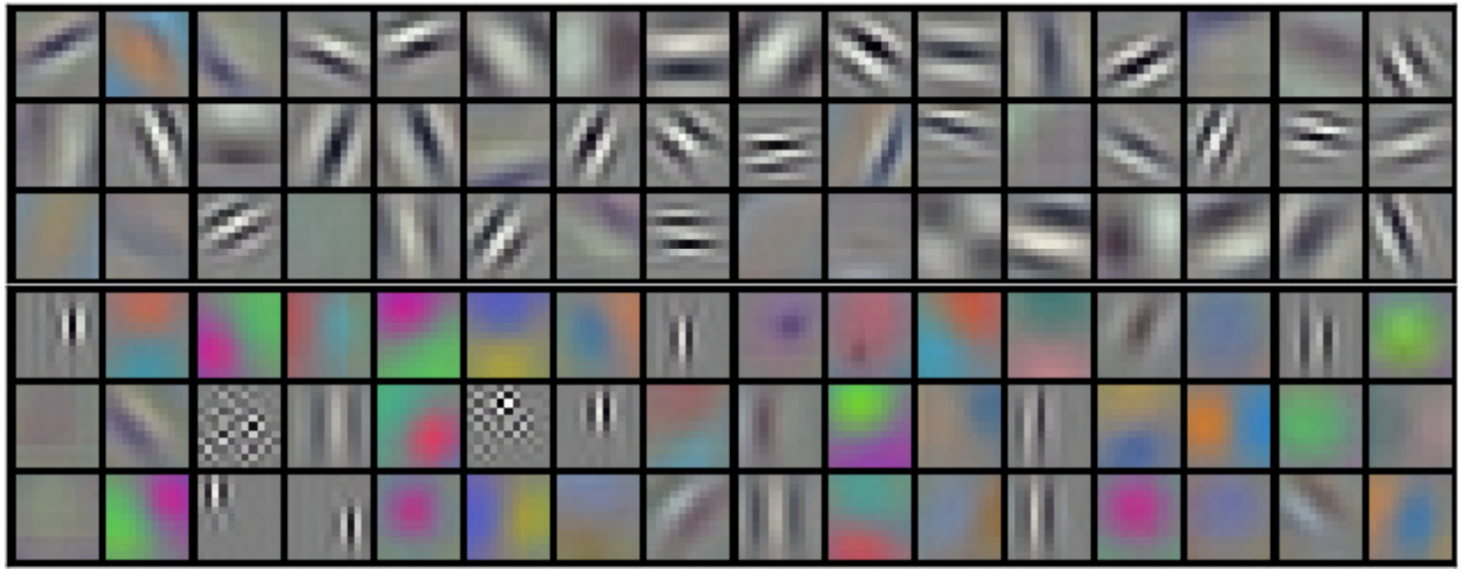
Redundancy in feature representations

ImageNet Challenge

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.



Learned convolution filters in the first layer



Symmetries in Medical Imaging

Problems with classical CNNs:

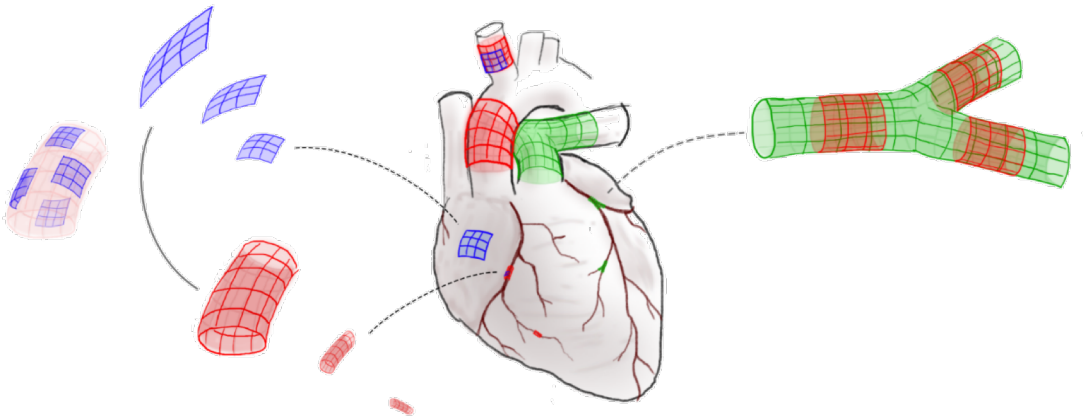
- No guarantee of equivariance (other then translation).
- Redundancy in feature representation.
- Artificially create extra data samples by data augmentation.

Solution:

Group convolutional neural networks: G-CNNs

Moreover, G-CNNs:

- Exploit symmetries in data.
- Weight sharing.
- Perform better by not having to spend valuable network capacity on learning geometric properties.



Theory: From NNs to CNNs to G-CNNs

NNs

CNNs

G-CNNs

Add geometric structure (local spatial coherence, weight sharing, …)

Adding more geometric structure (beyond translation equivariance)

Problem description

Given a training set $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{N}$ consisting of pairs of inputs x_i and desired outputs y_i, find a function $\Phi_{\underline{\omega}}$, parameterized by $\underline{\omega}$, that best maps each input to a desired output.

Here "best" is quantified by (local) minima of a loss:

$$\min_{\underline{\omega} \in \mathbb{R}^N} \; loss(\underline{\omega}, \mathcal{S})$$

computed by stochastic gradient descent (backward propagation via chain-law)

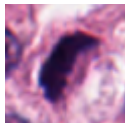
e.g. least squares

$$loss(\underline{\omega}, \mathcal{S}) = \tfrac{1}{2} \sum_{i=1}^{N} \|y_i - \Phi_{\underline{\omega}}(x_i)\|^2$$

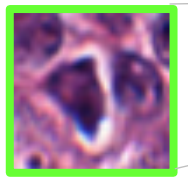
or (multi-class) cross-entropy

$$loss(\underline{\omega}, \mathcal{S}) = -\sum_{i=1}^{N} \sum_{c=1}^{N_{classes}} y_{i,c} \log(\Phi_{\underline{\omega}}(x_i)_c)$$

Training set

$$\mathcal{S} = \left\{ \begin{array}{cc} x_i & y_i \\ \left(\text{  } \right. & \left. \text{, 0: normal} \right) \\ \left(\text{  } \right. & \left. \text{, 0: normal} \right) \\ \left(\text{  } \right. & \left. \text{, 1: mitotic} \right) \\ \ldots & \end{array} \right\}$$

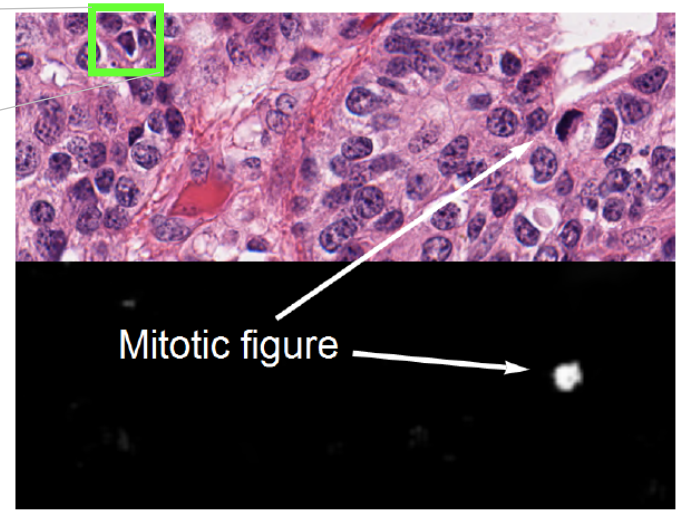
Testing



$\Phi_{\underline{\omega}}$

0.14

"probably a healthy cell"



Neural Network: $\Phi_{\underline{\omega}}$

- A Neural Network is a composition of operators:

$$\underline{x}^0 \overset{\Phi^1_{\omega^1}}{\mapsto} \underline{x}^1 \overset{\Phi^2_{\omega^2}}{\mapsto} \dots \overset{\Phi^L_{\omega^L}}{\mapsto} \underline{y}$$

- In which each operator ("layer") has the following form:

$$\underline{y} = \Phi_{\boldsymbol{\omega}}(\underline{x}) = \varphi\left(A_{\mathbf{w}}\underline{x} + \underline{b}\right)$$

$\underline{x} \in \mathcal{X}$, the input vector

$\underline{y} \in \mathcal{Y}$, the output vector

$A_{\mathbf{w}} : \mathcal{X} \to \mathcal{Y}$, a linear mapping parameterized by weights $\mathbf{w}$

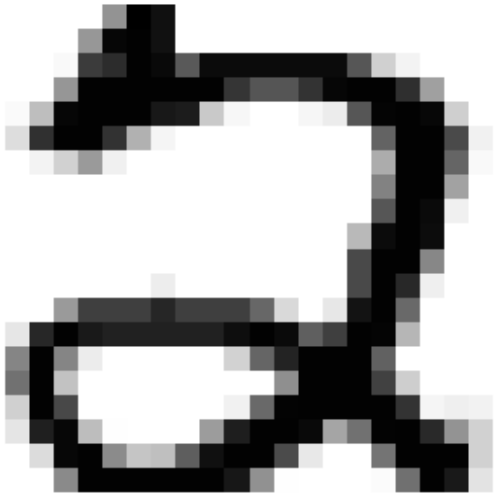
$\underline{b} \in \mathcal{Y}$, a bias term

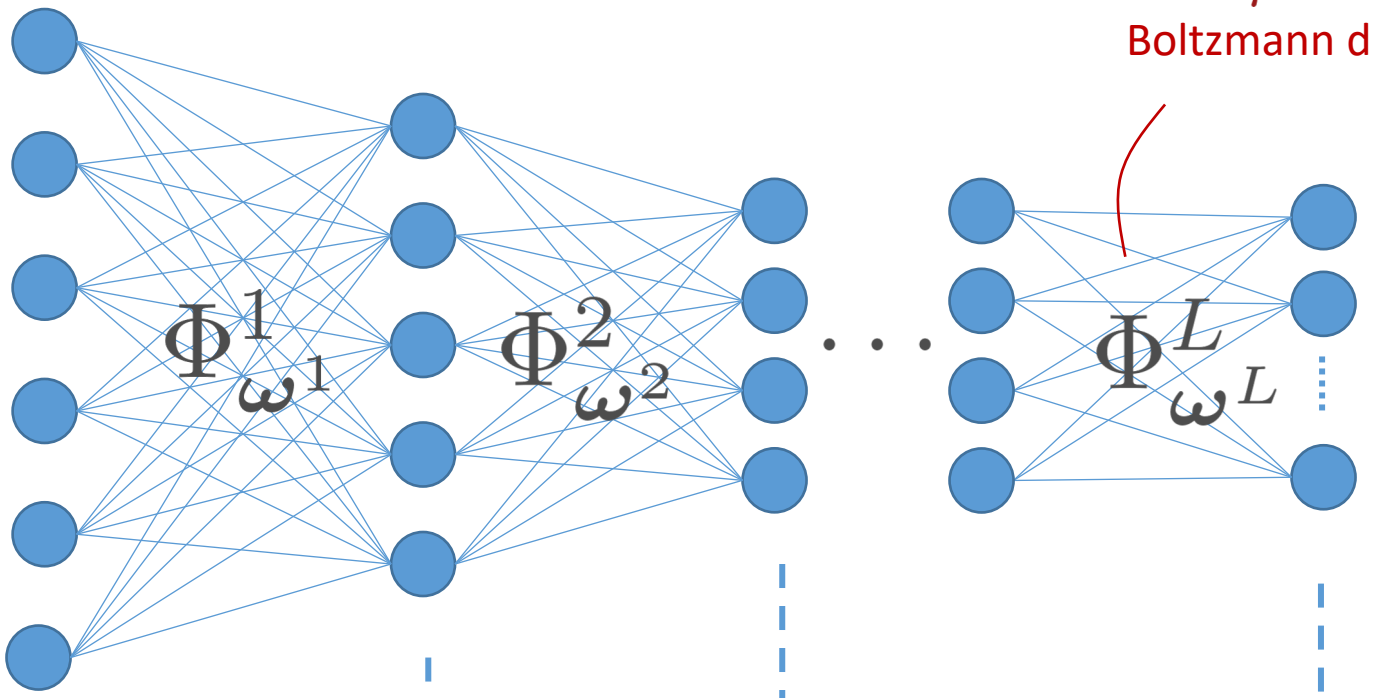
$\varphi : \mathbb{R} \to \mathbb{R}$ an activation function (applied element wise)

$\boldsymbol{\omega} = (\mathbf{w}, \underline{b})$ the trainable parameters

A classical neural network

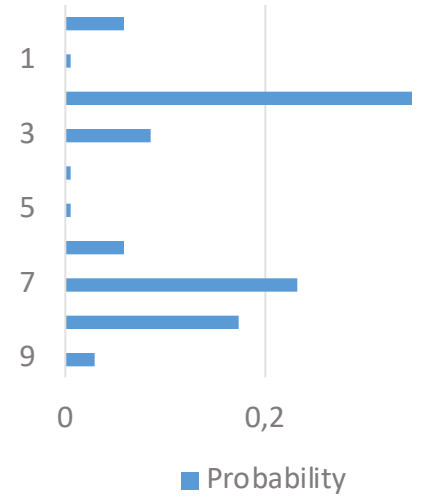
$\underline{x}^0 \in \mathcal{X} = ?$





With φ "soft-max" Boltzmann distr.

Class probability



2D function/image

$\mathcal{X} = \mathbb{L}_2(\mathbb{R}^2)$

2D array

$\mathcal{X} = \mathbb{R}^{28\times 28}$

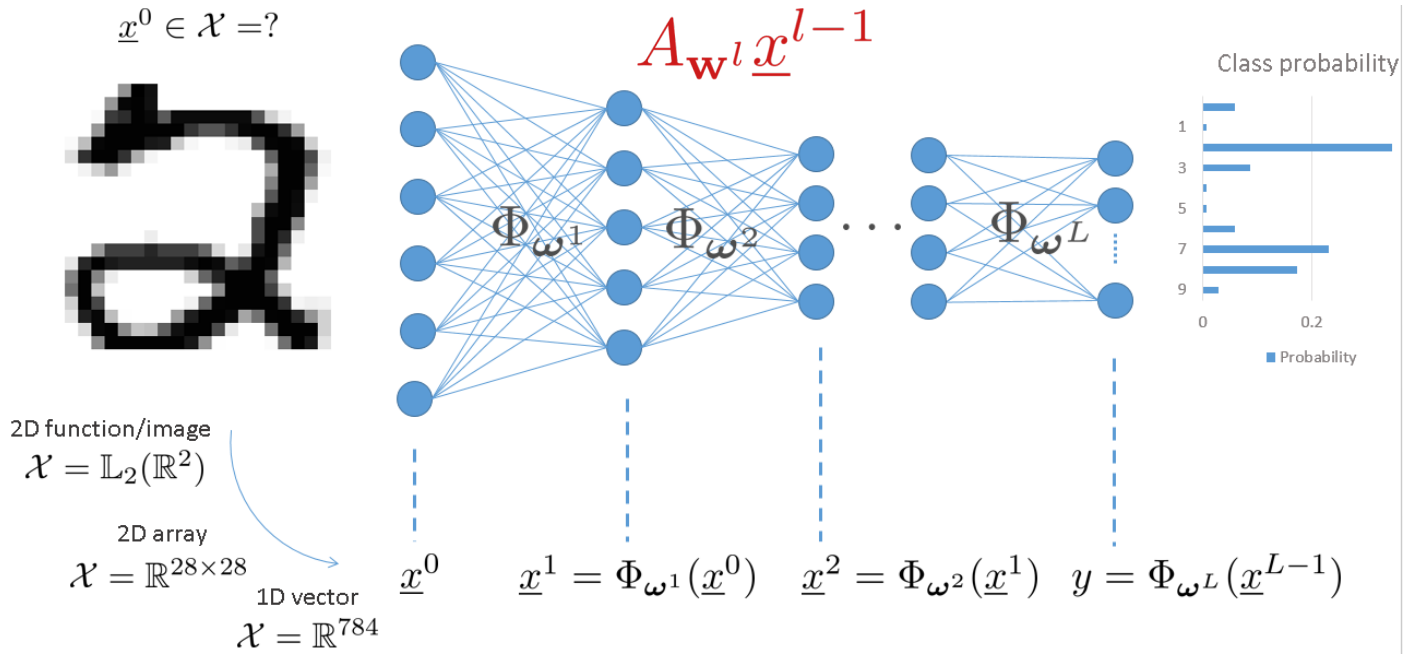
1D vector

$\mathcal{X} = \mathbb{R}^{784}$

$\underline{x}^0 \quad \underline{x}^1 = \Phi^2_{\omega^1}(\underline{x}^0) \quad \underline{x}^2 = \Phi^2_{\omega^2}(\underline{x}^1) \quad y = \Phi_{\omega^L}(\underline{x}^{L-1})$

A classical neural network

$$\underline{x}^l = \varphi\left(A_{\mathbf{w}^l}\,\underline{x}^{l-1} + \underline{b}^l\right)$$



$\underline{x}^{l-1} \in \mathbb{R}^{N^{l-1}}$, the input vector

$\underline{x}^l \in \mathbb{R}^{N^l}$, the output vector

$A_{\mathbf{w}} : \mathbb{R}^{N^l \times N^{l-1}}$, a linear mapping parameterized by weights $\mathbf{w} \in \mathbb{R}^n$

$\underline{b} \in \mathbb{R}^{N^l}$, a bias term

$\varphi : \mathbb{R} \to \mathbb{R}$ an activation function (applied element wise)

$\boldsymbol{\omega} = (\mathbf{w}, \underline{b})$ the trainable parameters

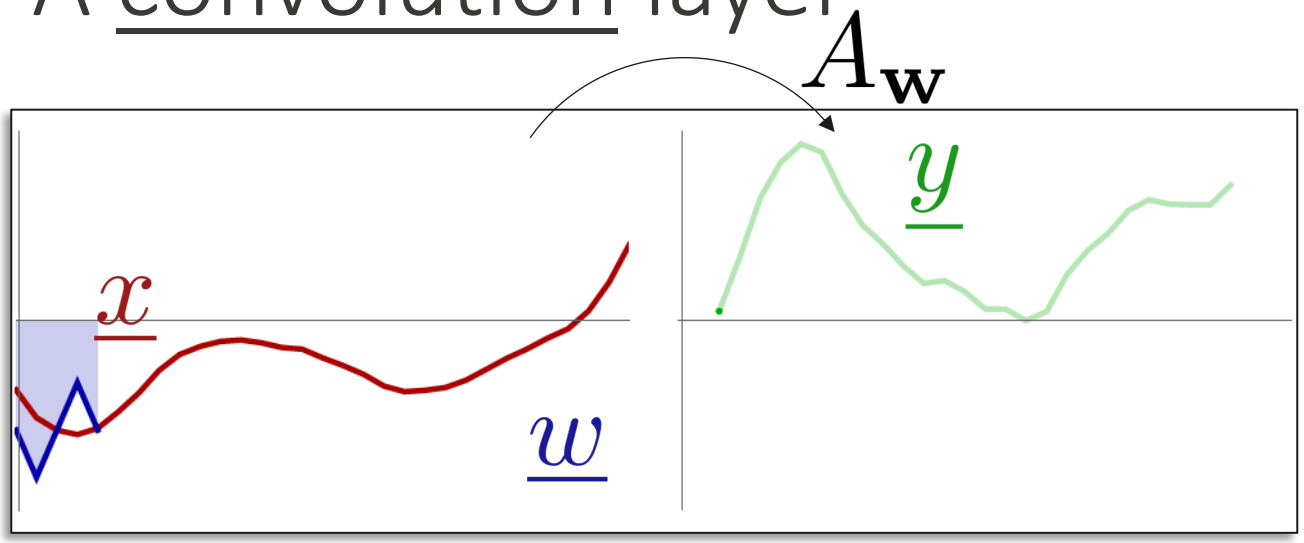
A fully connected neural layer

-Too many degrees of freedom
-Does not exploit structure in data

$$\underline{y} \qquad A_{\mathbf{w}} \qquad \underline{x} \qquad \underline{b}$$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \end{pmatrix} = \varphi \left(\begin{pmatrix} w_1^1 & w_1^2 & w_1^3 & w_1^4 & w_1^5 & \dots \\ w_2^1 & w_2^2 & w_2^3 & w_2^4 & w_2^5 & \dots \\ w_3^1 & w_3^2 & w_3^3 & w_3^4 & w_3^5 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ \vdots \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \end{pmatrix} \right)$$

A convolution layer



+ Localized transformations
+ Shift equivariance
+ Sparsification of the linear operator
+ Weight sharing

$$\underline{y} = A_{\mathbf{w}}\,\underline{x} + \underline{b}$$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \end{pmatrix} = \varphi\left(\begin{pmatrix} w^1 & w^2 & w^3 & 0 & 0 & \dots \\ 0 & w_1 & w^2 & w^3 & 0 & \dots \\ 0 & 0 & w^1 & w^2 & w^3 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ \vdots \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \end{pmatrix} \right)$$

Theory: From NNs to CNNs to G-CNNs

NNs

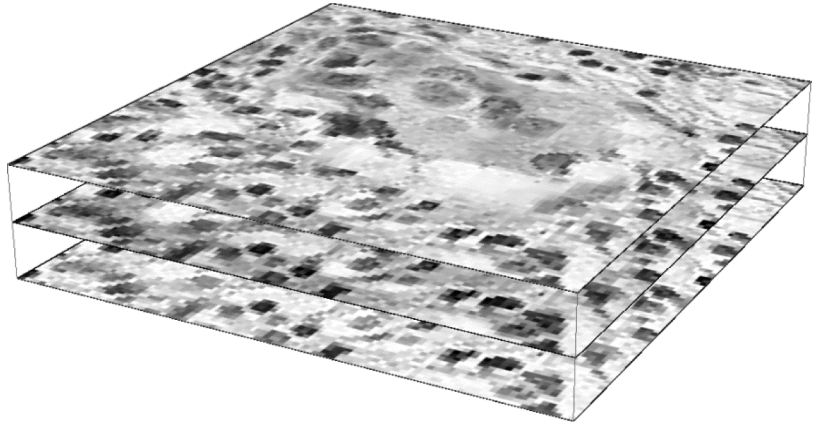
CNNs

G-CNNs

Input and output spaces

Input vector

$$\underline{f}^{l-1} \in \mathcal{X} = (\mathbb{L}_2(\mathbb{R}^2))^{N_c^{l-1}}$$

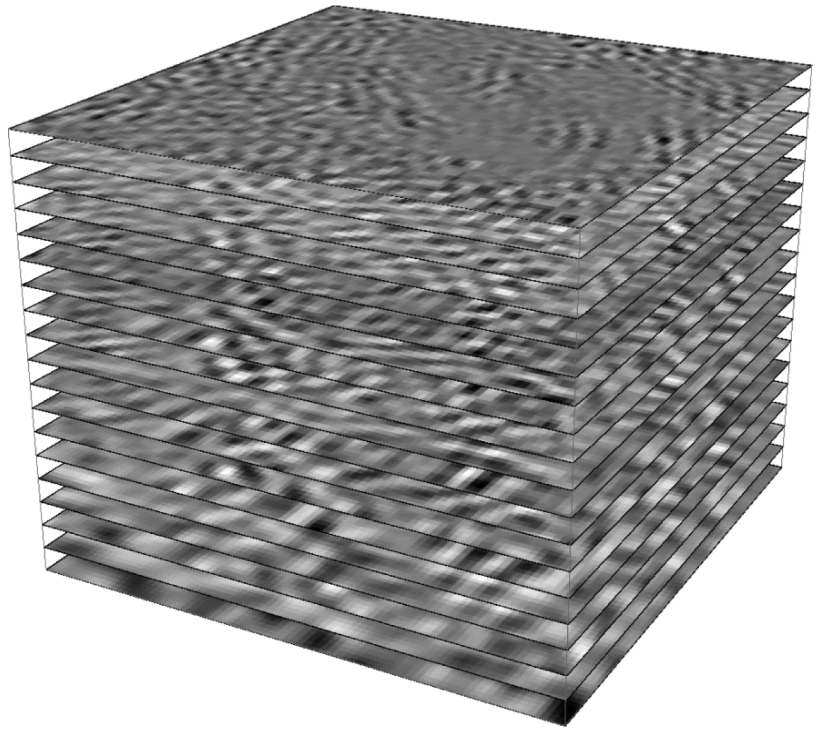


$$\Phi_{\omega^l}$$

Convolution layer

Output vector

$$\underline{f}^{l} \in \mathcal{Y} = (\mathbb{L}_2(\mathbb{R}^2))^{N_c^{l}}$$



Convolution layer (cross-correlation layer)

Translation by x

$$(k \star f)(\mathbf{x}) = (\mathcal{T}_{\mathbf{x}} k, f)_{\mathbb{L}_2(\mathbb{R}^2)} \qquad \text{with } k, f \in \mathbb{L}_2(\mathbb{R}^2)$$

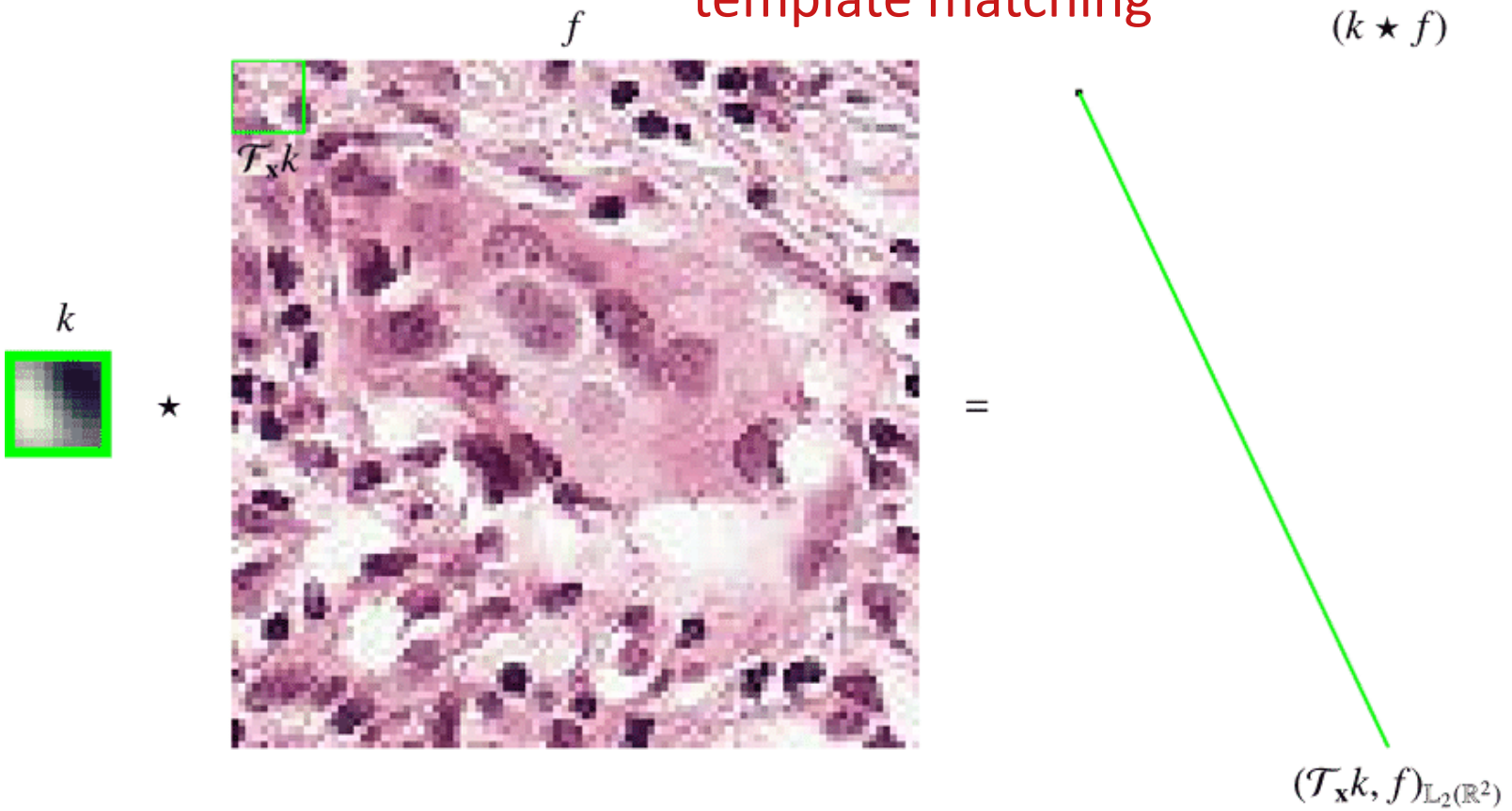
Convolution layer (cross-correlation layer)

Translation by x

$$(k \star f)(\mathbf{x}) = (\mathcal{T}_{\mathbf{x}} k, f)_{\mathbb{L}_2(\mathbb{R}^2)} = \int_{\mathbb{R}^2} k(\mathbf{x}' - \mathbf{x}) f(\mathbf{x}') \mathrm{d}\mathbf{x}',$$

note $(k \star f) = (\check{k} * f)$, with $\check{k}(\mathbf{x}) = k(-\mathbf{x})$

template matching



Convolution layers on $\mathbb{R}^n$

$(\mathbb{L}_2(\mathbb{R}^2))^{N_c^l}$ $(\mathbb{L}_2(\mathbb{R}^2))^{N_c^{l-1}}$

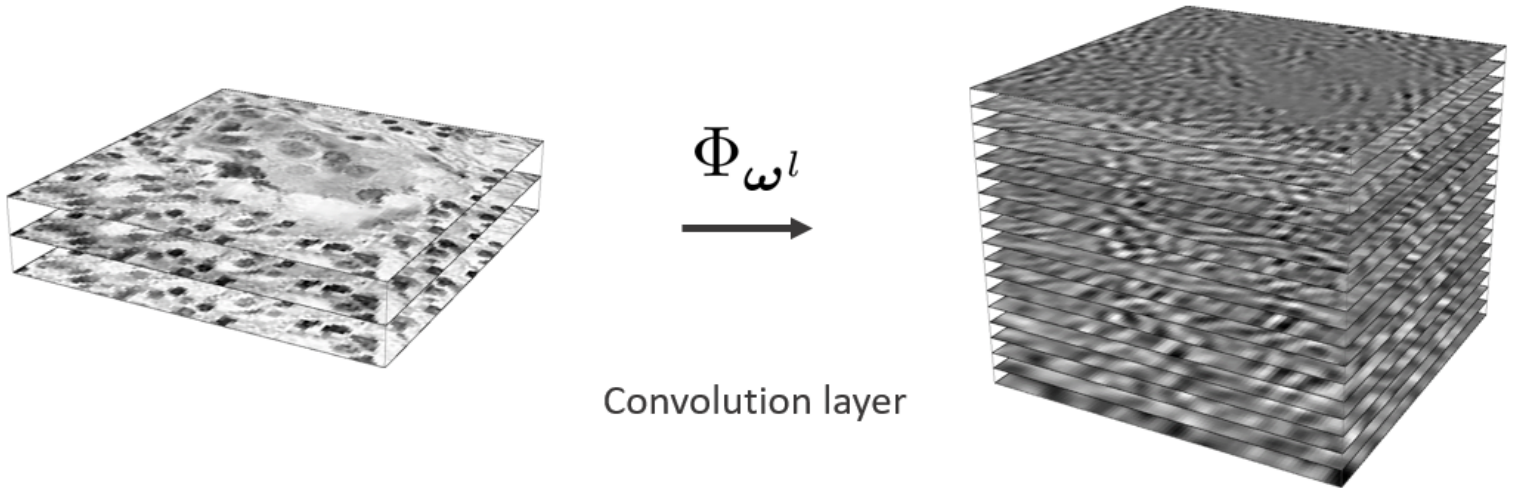
$$\underline{f}^l = \Phi_{\boldsymbol{\omega}^l}(\underline{f}^{l-1}) = \varphi\left(A^l_{\mathbf{w}^l}\underline{f}^{l-1} + \underline{b}^l\right)$$

A linear mapping

$$A^l_{\mathbf{w}^l}\underline{f}^{l-1} = \mathbf{w}^l \star \underline{f}^{l-1}$$

parameterized by a collection of convolution filters

$$\mathbf{w}^l \in (\mathbb{L}_2(\mathbb{R}^2))^{N_c^l \times N_c^{l-1}}$$



A typical CNN architecture

LeCun et al. 1989-1998

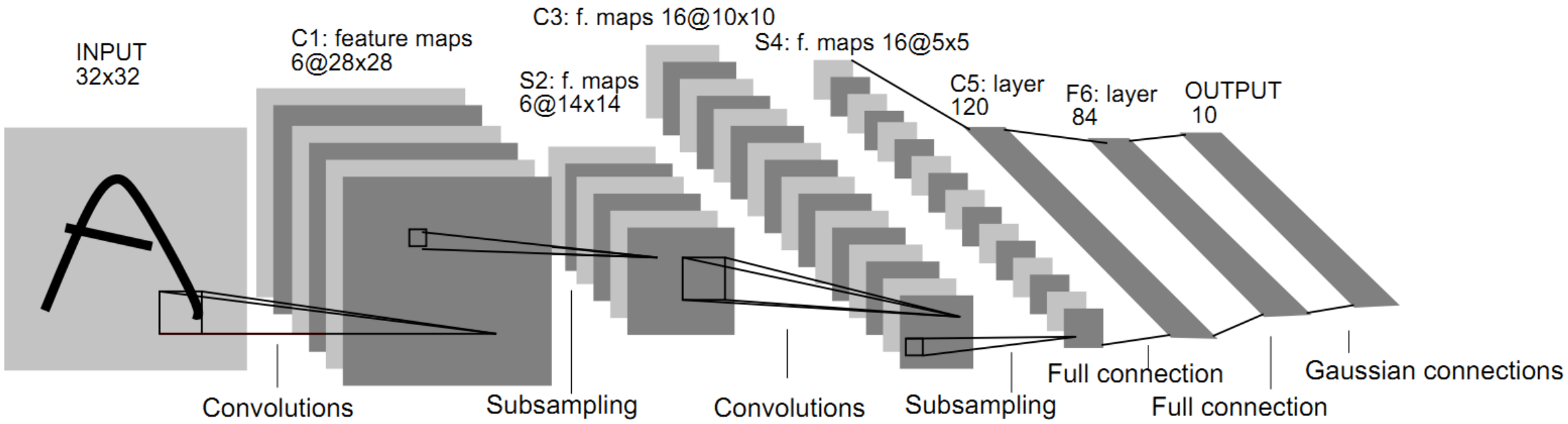


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Theory: From NNs to CNNs to G-CNNs

NNs

CNNs

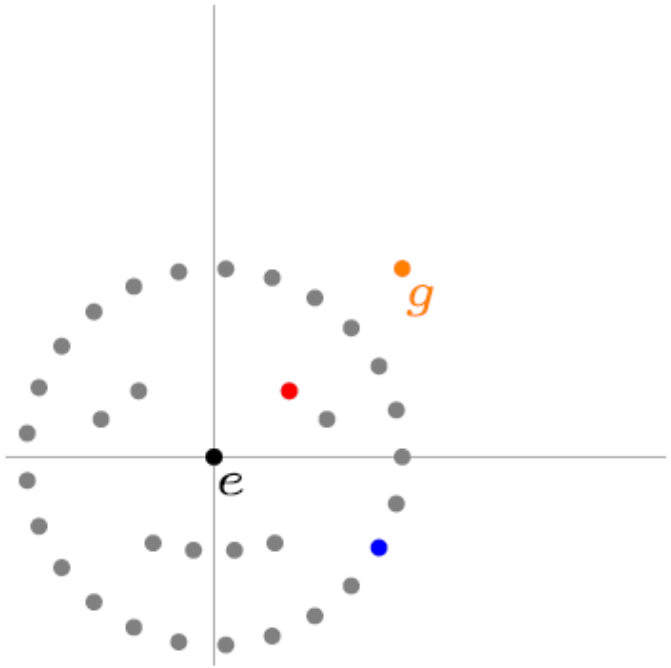
G-CNNs

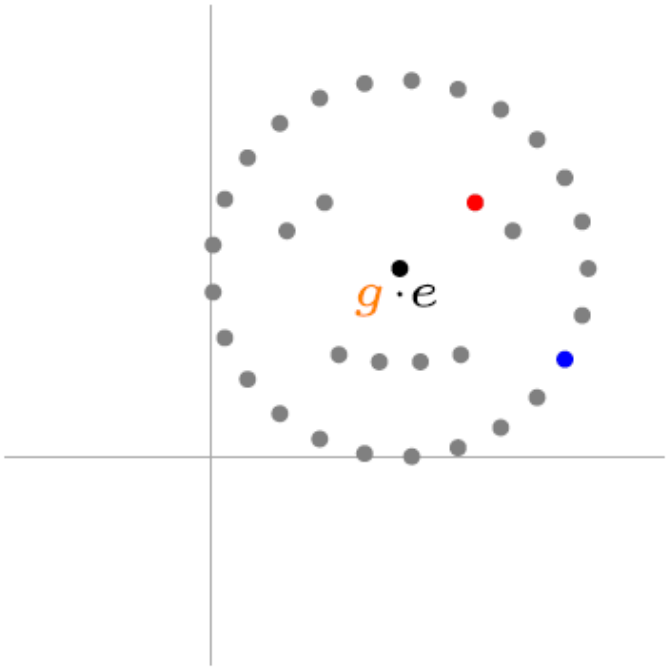
The translation group

The translation group consists of all possible translations in $\mathbb{R}^2$ and is equipped with the group product and group inverse:

$$\begin{aligned} g \cdot g' &= \mathbf{x} + \mathbf{x}' \\ g^{-1} &= -\mathbf{x}. \end{aligned}$$

with $g = \mathbf{x}, g' = \mathbf{x}' \in \mathbb{R}^2$.

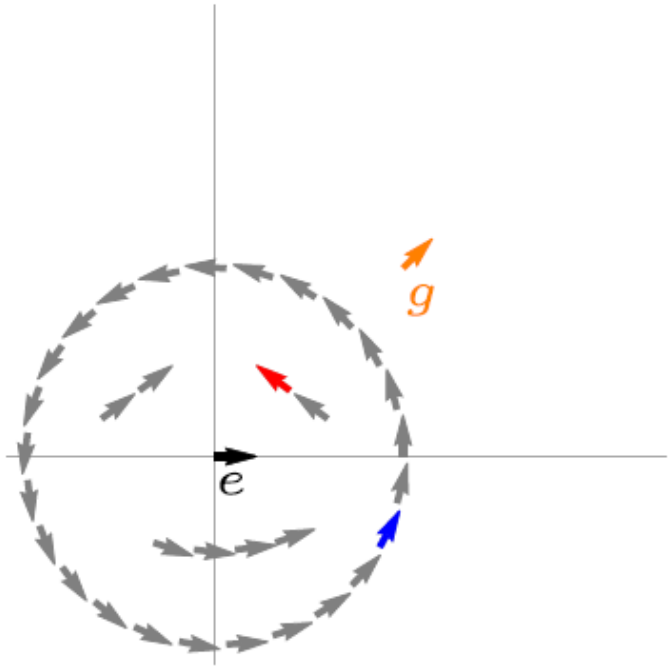


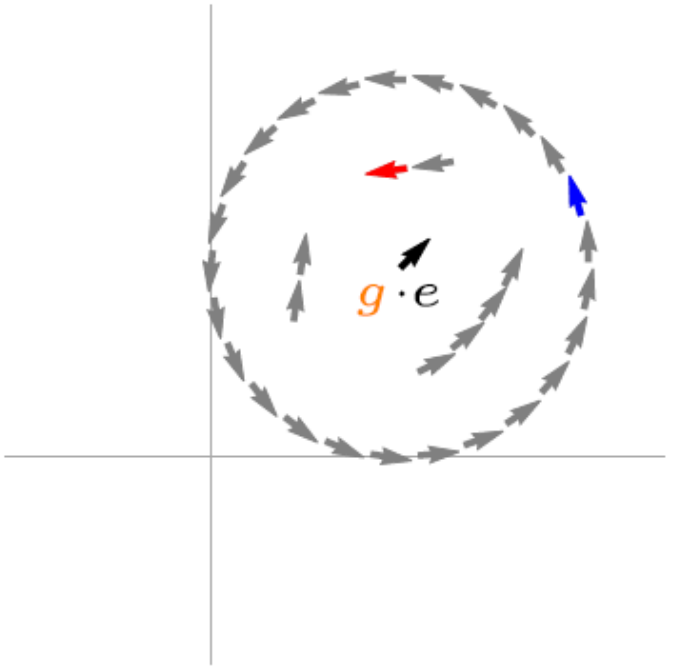


The roto-translation group SE(2)

The group $SE(2)$ consists of the coupled space $\mathbb{R}^2 \rtimes S^1$ of positions (translations) in $\mathbb{R}^2$, and orientations (rotations) S^1, and is equipped with the group product and group inverse:

$$\begin{aligned} g \cdot g' &= (\mathbf{x}, \mathbf{R}_\theta) \cdot (\mathbf{x}', \mathbf{R}_{\theta'}) = (\mathbf{R}_\theta \mathbf{x}' + \mathbf{x}, \mathbf{R}_{\theta+\theta'}) \\ g^{-1} &= (-\mathbf{R}_\theta^{-1}\mathbf{x}, \mathbf{R}_\theta^{-1}). \end{aligned}$$





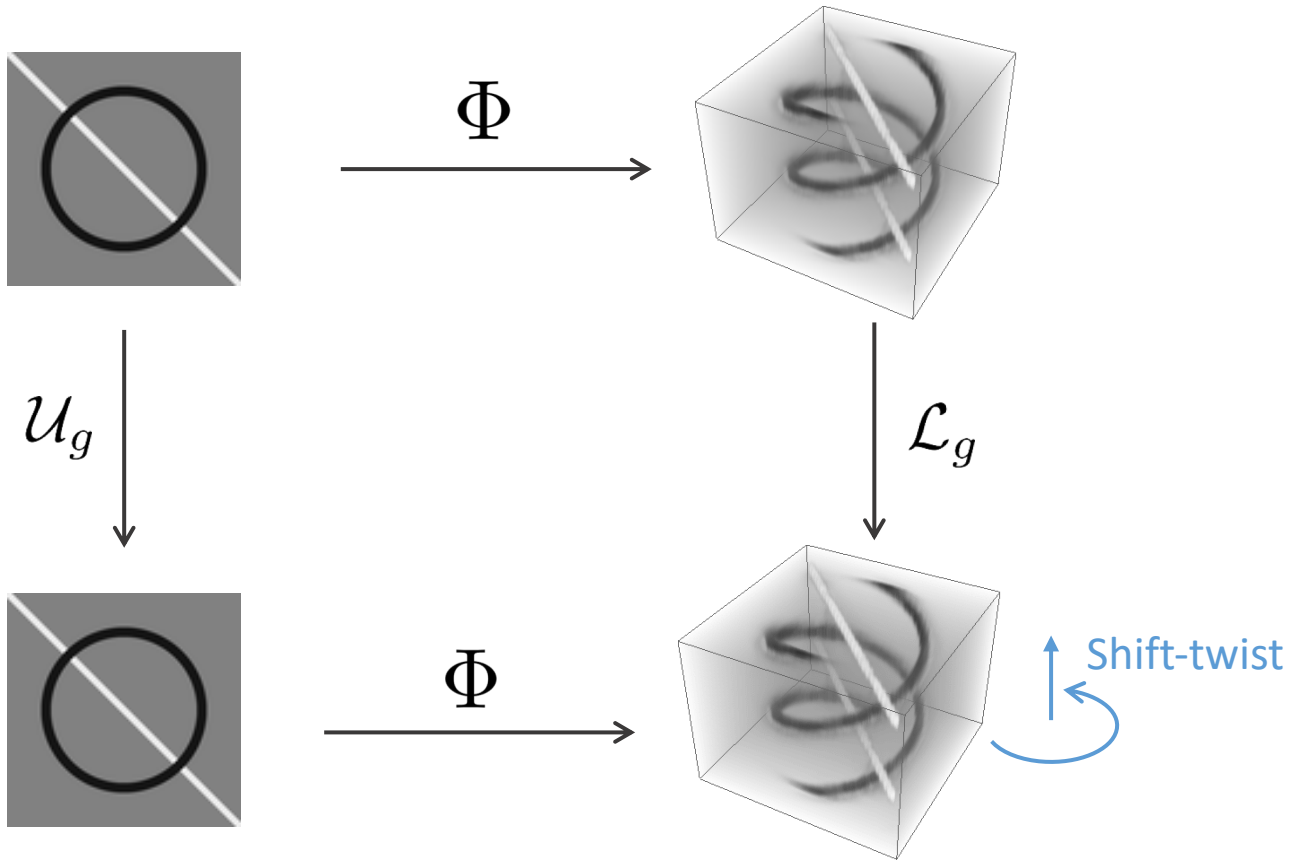
Group Representations of SE(2)

The representation of SE(2) on $f \in \mathbb{L}_2(\mathbb{R}^2)$

$$(\mathcal{U}_g f)(\mathbf{y}) = f(g^{-1} \odot \mathbf{y}) = f(\mathbf{R}_\theta^{-1}(\mathbf{y} - \mathbf{x}))$$

The representation of SE(2) on $F \in \mathbb{L}_2(SE(2))$

$$(\mathcal{L}_g F)(\mathbf{y}) = F(g^{-1} \cdot h) = F(\mathbf{R}_\theta^{-1}(\mathbf{y} - \mathbf{x}), \alpha - \theta)$$

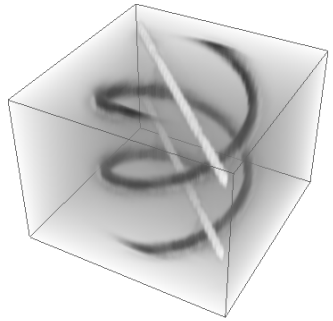


Group correlation layers

Lifting layer:

$$(k \star f)(g) := (\mathcal{U}_g k, f)_{\mathbb{L}_2(\mathbb{R}^2)}$$

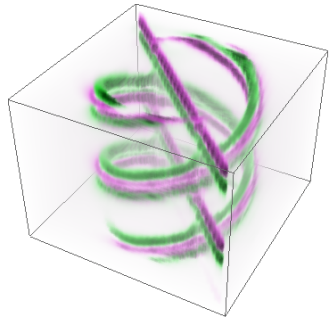
Φ^1



G-correlation layer:

$$(K \star F)(g) := (\mathcal{L}_g K, F)_{\mathbb{L}_2(SE(2)}$$

Φ^2



Group correlation layers

SE(2) equivariance

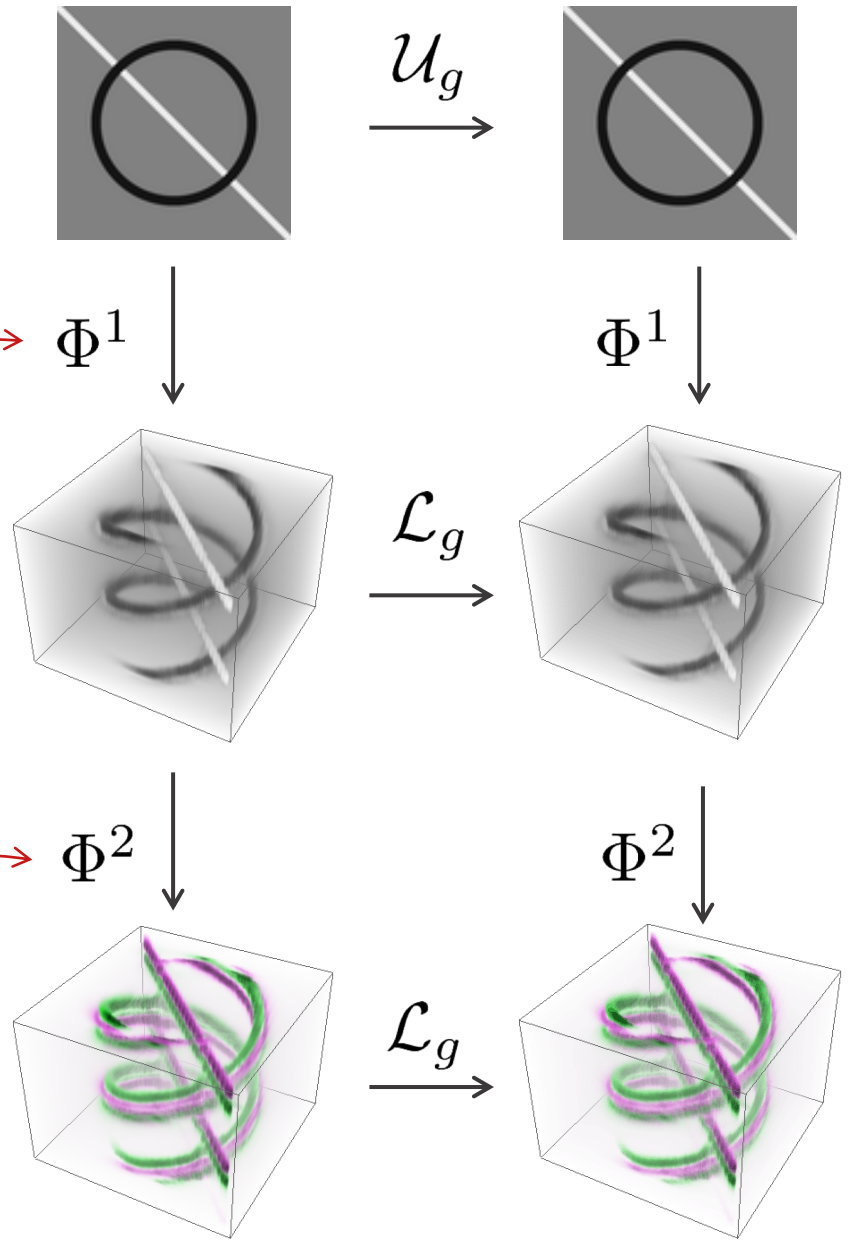
Lifting layer:

$$(k \star f)(g) := (\mathcal{U}_g k, f)_{\mathbb{L}_2(\mathbb{R}^2)}$$

G-correlation layer:

$$(K \star F)(g) := (\mathcal{L}_g K, F)_{\mathbb{L}_2(SE(2))}$$

Note: $K \star F = \check{K} * F$ with $\check{K}(g) = K(g^{-1})$



Theorem on Equivariant Linear Operators motivating the

G-CNN Design

Thm. Let X, Y be homogeneous spaces. Let Lie group G act transitively on Y. Let μ_X and μ_Y be Radon measures on X resp. Y s.t.

$$\exists_{m\in\mathbb{L}_1(G)}\forall_{g\in G}\forall_{x\in X} \;:\; \frac{\mathrm{d}\mu_X(g^{-1}x)}{\mathrm{d}\mu_X(x)} = m(g)$$

Let $K : \mathbb{L}_2(X, \mathrm{d}\mu_X) \to \mathbb{L}_2(Y, \mathrm{d}\mu_Y)$ be bounded linear s.t. $\sup_{\|f\|=1} \|Kf\|_{\mathbb{L}_\infty} < \infty$.

Dunford-Pettis: $\exists!\, _{k\in\mathbb{L}_1(Y\times X)}\forall_{f\in\mathbb{L}_2(X,\mathrm{d}\mu_X)}\, \forall_{y\in Y} : (Kf)(y) = \int_X k(y,x)f(x)\,\mathrm{d}\mu_X(x).$

Let A, B be linear left-regular reps. of G on X, Y s.t. $\forall_{g\in G} \;:\; K\circ A_g = B_g \circ K$.

Then $\exists_{y_0\in Y}\forall_{y\in Y}\exists_{g_y\in G} \;:\; y = g_y\; y_0$ and

$$\exists!\, _{k_0\in\mathbb{L}_1(X)}\forall_{f\in\mathbb{L}_2(X,\mathrm{d}\mu_X)}\; '\forall'_{y\in Y} : (Kf)(y) = \int_X m(g_y)\; k_0(g_y^{-1}x) f(x)\,\mathrm{d}\mu_X(x)$$

$$\text{with } \forall_{h\,\in\,\mathrm{Stab}_G(y_0)} \;:\; k_0(x) = m(h)\, k_0(hx).$$

Special cases :

- $X = Y = \mathbb{L}_2(\mathbb{R}^d)$ & $G = SE(d)$. Then $g \odot \mathbf{x} = (\mathbf{b}, \mathbf{R})\mathbf{x} = \mathbf{R}\mathbf{x} + \mathbf{b}$.
  Then $Kf = k_0 * f$ with k_0 isotropic.

- $X = \mathbb{L}_2(\mathbb{R}^d)$ & $Y = \mathbb{L}_2(G)$ & $G = SIM(d)$: $g = (\mathbf{x}, \mathbf{R}, a)$, $m(g) = a^{-d}$,

$$Kf(g) = a^{-\frac{d}{2}} \int_{\mathbb{R}^d} \overline{\psi}(a^{-1}\mathbf{R}^{-1}(\mathbf{x}' - \mathbf{x}))f(\mathbf{x}')\,\mathrm{d}\mathbf{x}' \text{ with } \psi = a^{\frac{d}{2}}\overline{k}_0.$$

- $X = \mathbb{L}_2(\mathbb{R}^d)$ & $Y = \mathbb{L}_2(G)$ & $G = SE(d)$: $Kf = (\mathcal{U}_g\psi, f)$ with $\psi = \overline{k}_0$.

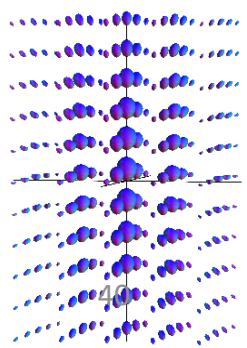
- $X = Y = \mathbb{L}_2(G)$ & $G = SE(3)$:

$$Kf(g) = (k * f)(g) = \int_G k(h^{-1}g)f(h)\,\mathrm{d}\mu_G(h).$$

- $X = Y = \mathbb{L}_2(\mathbb{R}^3 \times S^2)$ & $G = SE(d)$: $m = 1$, $\mathbf{y}_0 = (\mathbf{0}, \mathbf{a})$ and

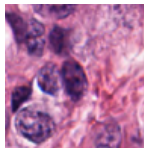
$$Kf(\mathbf{x}, \mathbf{n}) = (k_0 * f)(\mathbf{x}, \mathbf{n}) = \int_{\mathbb{R}^3} \int_{S^2} k_0(\mathbf{R}_{\mathbf{n}'}^T(\mathbf{x} - \mathbf{x}'), \mathbf{R}_{\mathbf{n}'}^T\mathbf{n})f(\mathbf{x}', \mathbf{n}')\,\mathrm{d}\mathbf{x}'\mathrm{d}\sigma(\mathbf{n}').$$

$$k_0(\mathbf{R}_{\mathbf{a},\alpha}\mathbf{x}, \mathbf{R}_{\mathbf{a},\alpha}\mathbf{n}) = k_0(\mathbf{x}, \mathbf{n})$$



⇒ G-CNN design for rotation invariant patch classification

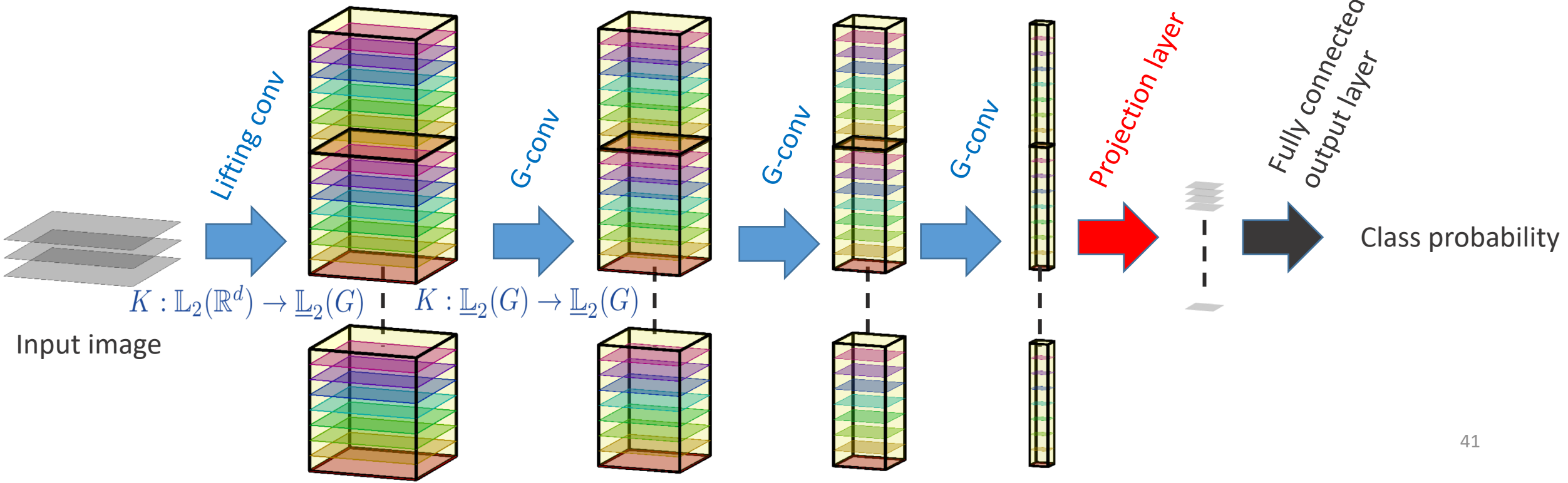
Bekkers et al.
MICCAI 2018
ICIAR 2014 (2 layers)



"normal" (0) vs "mitotic" (1)

Max-pooling over rotations only
Equivariance. Better choices…

Thm. ⇒ equivariant linear operator design !
(in practice interleafed ReLu: not affecting equiv.)



Steerable Implementations possible via Fourier Transform on the homogeneous space of positions and orientations:

$$\mathbb{R}^d \rtimes S^{d-1} := G/H = SE(d)/(\{\mathbf{0}\} \times SO(d-1))$$

i.e. identify $(\mathbf{x}, \mathbf{n}) \leftrightarrow (\mathbf{x}, \mathbf{R_n})$ with $\mathbf{R_n}$ *any* rotation mapping $\mathbf{a}$ to orientation $\mathbf{n}$

d=2: $\mathbb{R}^2 \rtimes S^1 = SE(2)$: $\quad f_1 *_G f_2 = \mathcal{F}_G^{-1}(\mathcal{F}_G f_1 \circ \mathcal{F}_G f_2)$

Chirikjian & Kyatkin,
PhD Theses: E.M.Franken, L.Siffre

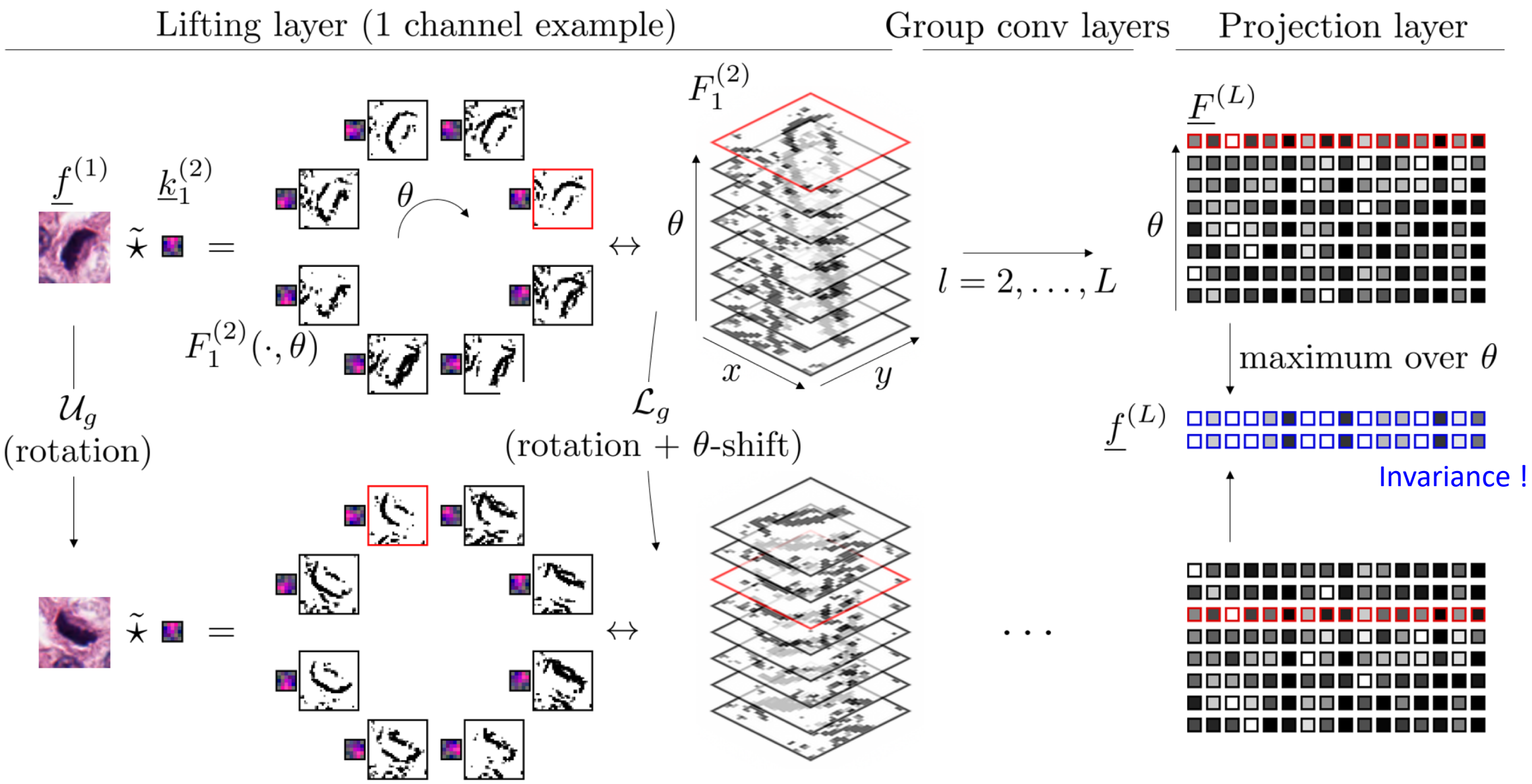
d=3: $\mathbb{R}^3 \rtimes S^2 \equiv SE(3)/(\{0\} \times SO(2))$:

$$f_1 *_{G/H} f_2 = \mathcal{F}_{G/H}^{-1}(\mathcal{F}_{G/H} f_1 \circ \mathcal{F}_{G/H} f_2)$$

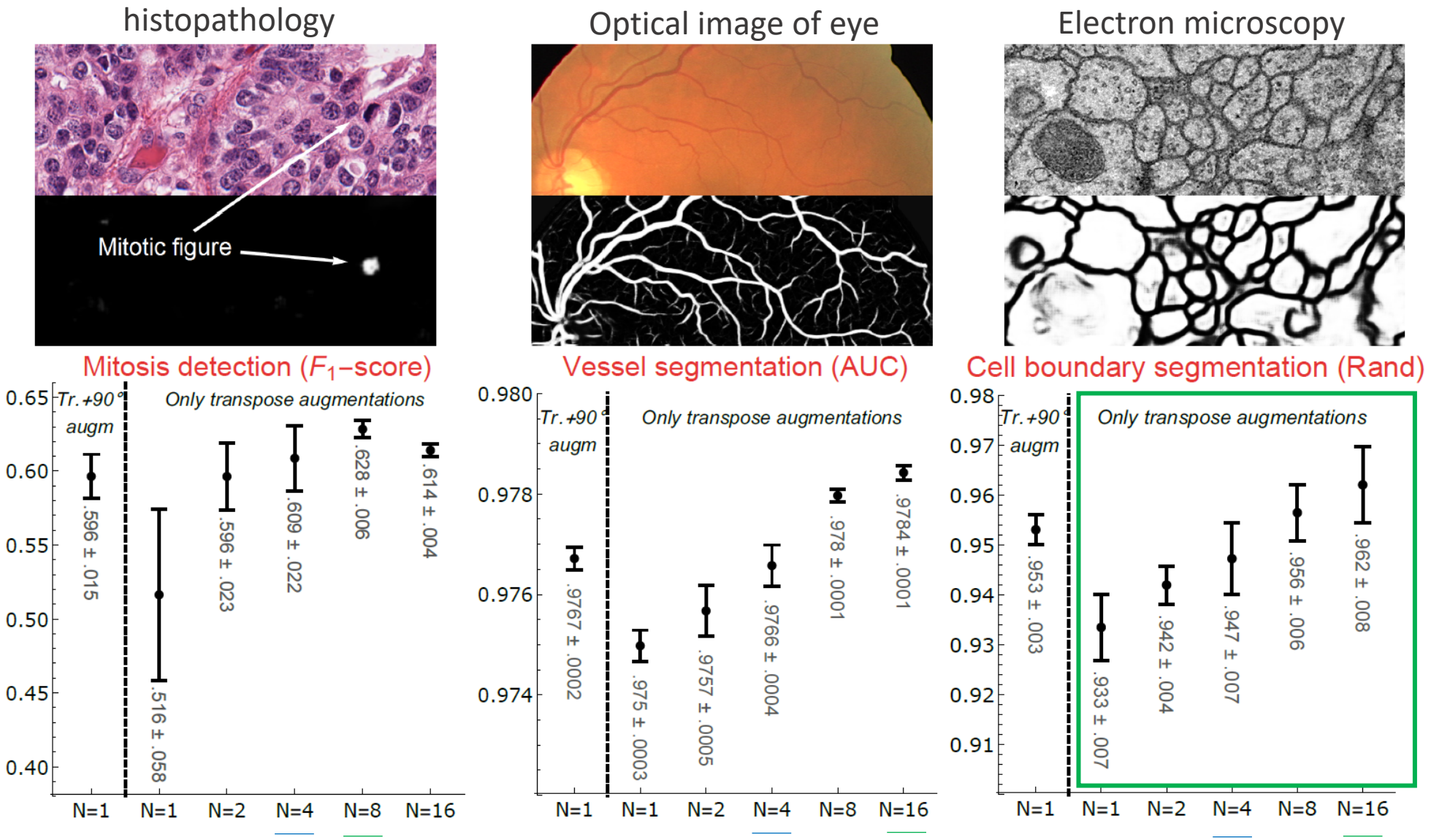
Duits & Bekkers & Mashtakov
Entropy 2019.
SI: In honor of 250 years Fourier.

Results

Example (rotation equivariance and invariance)



Results



Same capacity

But no waist as N increases.

Fig. 2: Top row: Crop outs of images of the three tasks with the class probabilities generated by our method. Bottom row: Mean results (±1 std. dev.).

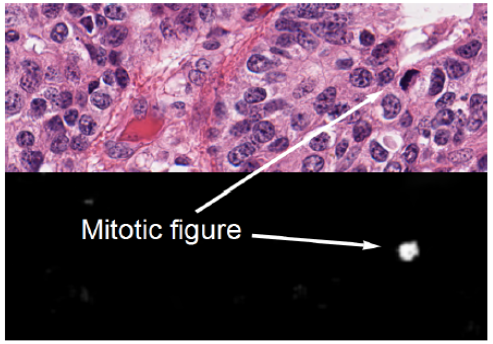
Conclusion

Conclusion

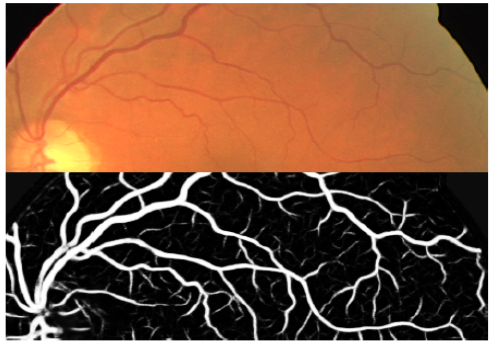
- Roto-translation ($SE(2)$) covariant CNNs
  - Lifting layer
  - G-conv layer (interleafed with ReLU)
  - Projection layer
- Increased performance compared to 2D benchmarks that rely on rotation augmentation
- Code + demos are on GitHub
  - github.com/tueimage/se2cnn
  - G-conv layers can replace standard conv2d layers
  - Or combined with standard 2D layers (see MNIST demo)

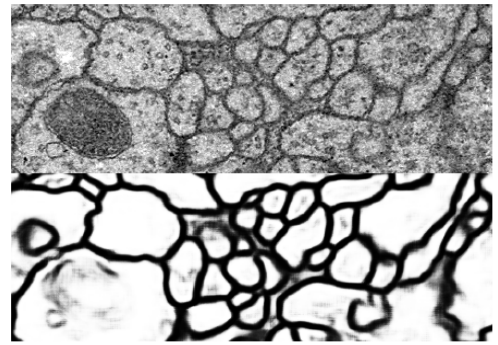
- PDE Geometric Processing on $SE(d)$.
  2005–ongoing...
  Machine Learning also on $SE(d)$!
  2014–ongoing... Next...

Better results by deeper-layers and multiple orientations!



Mitosis detection (F_1–score)


Vessel segmentation (AUC)


Cell boundary segmentation (Rand)