

Geometric Learning with Lie Groups and PDEs

Outline

- TU/e Geometric Learning & Differential Geometry group
- PDE-based Group equivariant Neural Networks
 - Motivation
 - Construction
 - Experiments
- LieTorch software package

TU/e > Geometric Learning and Differential Geometry Group



prof.dr.ir
Remco Duits



dr.ir.
Bart Smets

PhD candidates



Gijs Bellaard
MSc.



Nicky van den Berg
MSc.

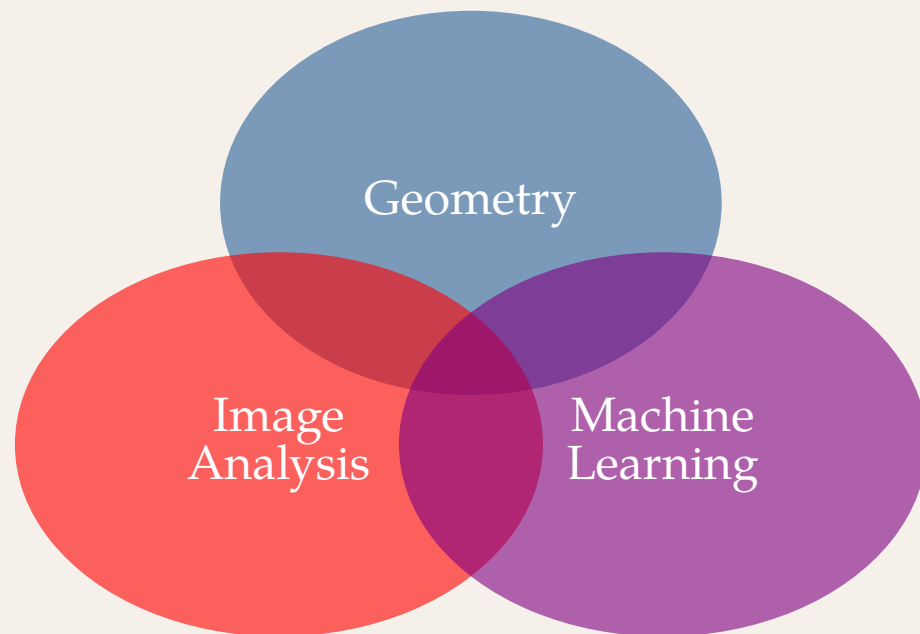


Leanne Vis
MSc.



Finn Sherry
MSc.

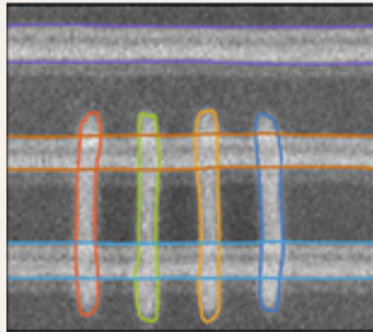
Active Fields



Running projects

SEM image analysis
for semiconductor
manufacturing

ASML

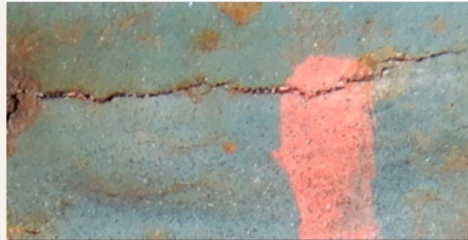


Crack detection for
automated bridge inspection



Rijkswaterstaat
Ministerie van Infrastructuur en Milieu

ProRail



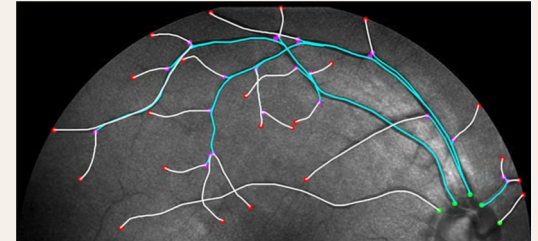
Retinal vessel
segmentation & tracking



**Maastricht
University**

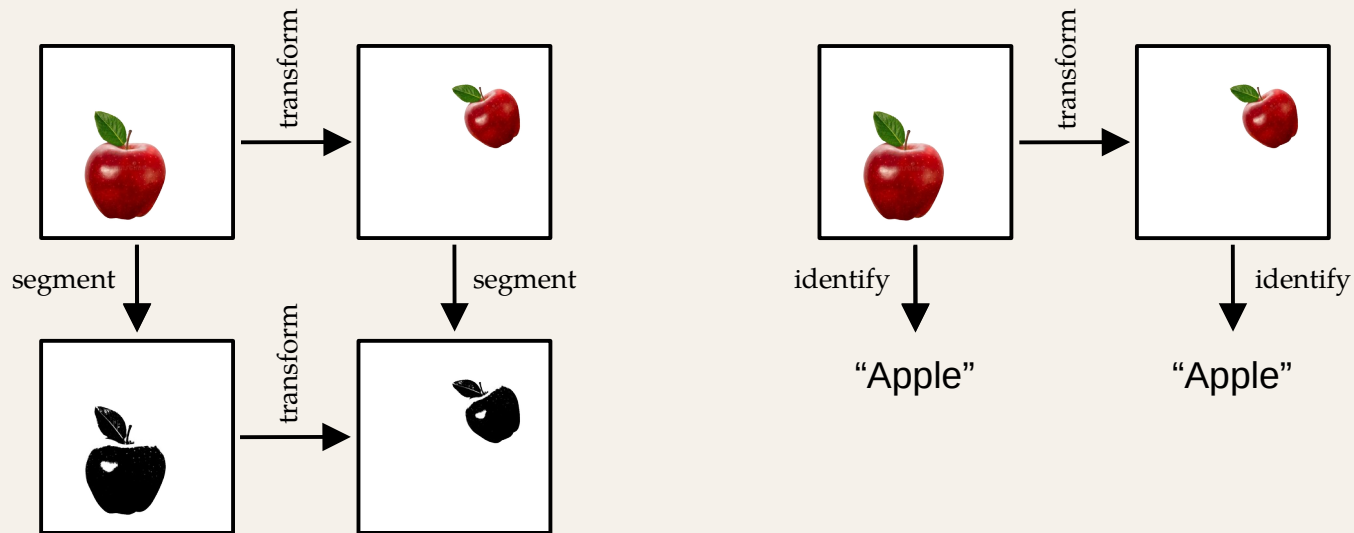


Maastricht UMC+



Why geometry?

- Equivariance & invariance



- Little available data
- Interpretability

PDE-based Group Equivariant CNNs (PDE-G-CNNs)

[1] Smets B.M.N., Portegies J., Bekkers E.J., Duits R. **PDE-Based Group Equivariant Convolutional Neural Networks**. J Math Imaging Vis 65, 209–239 (2022).

<https://doi.org/10.1007/s10851-022-01114-x>

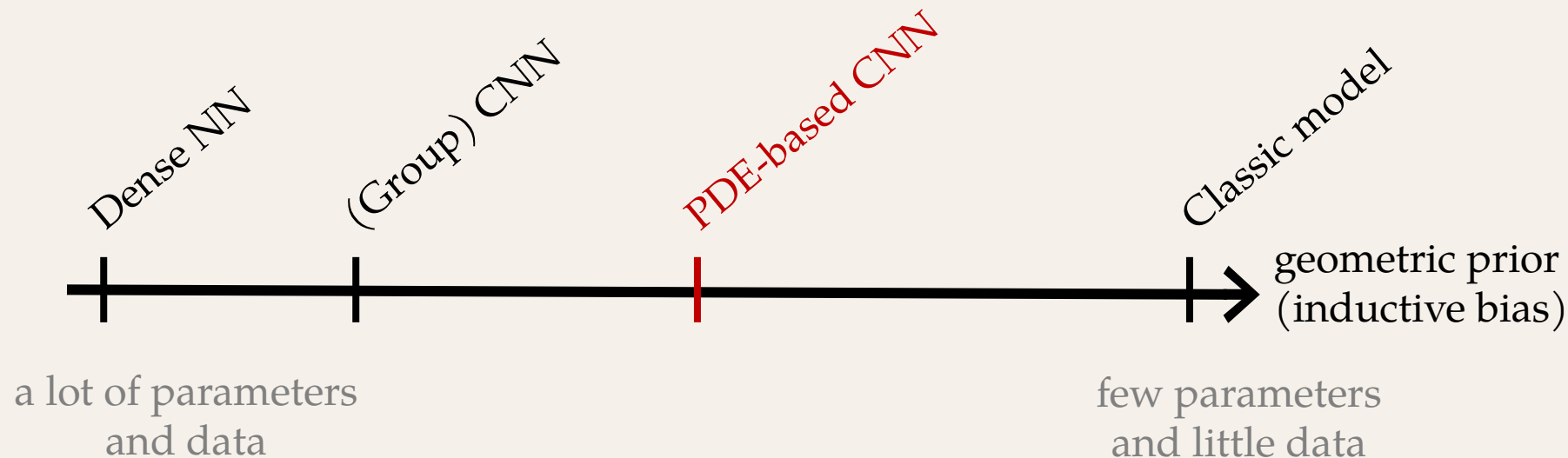
[2] Bellaard G., Bon D.L.J., Pai G., Smets B.M.N., Duits R. **Analysis of (sub-)Riemannian PDE-G-CNNs**. J Math Imaging Vis (2023).

<https://doi.org/10.1007/s10851-023-01147-w>

[3] Pai, G. Bellaard G., Smets B.M.N., Duits R. (2023). **Functional Properties of PDE-based Group Equivariant Convolutional Neural Networks**. In: Nielsen, F., Barbaresco, F. (eds) Geometric Science of Information. GSI 2023. Lecture Notes in Computer Science, vol 14071. Springer, Cham.

https://doi.org/10.1007/978-3-031-38271-0_7

Geometric prior

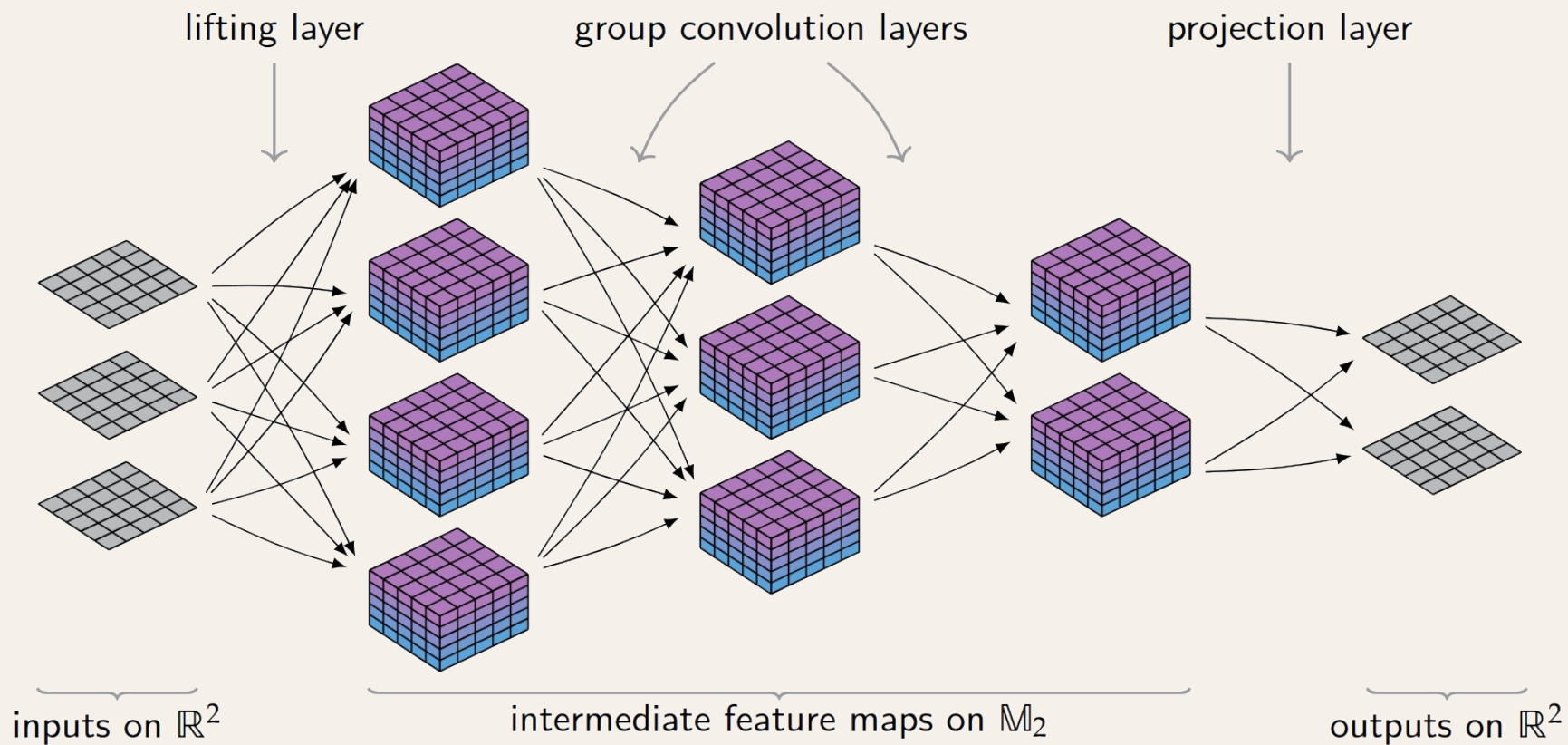


Starting point: Group CNN

- neural network + translation equivariance = CNN
 $G = T(2)$ (translations in 2 dimensions)
- neural network + roto-translation equivariance = SE(2)-CNN
 $G = SE(2)$ (rotations and translation in 2 dimensions)

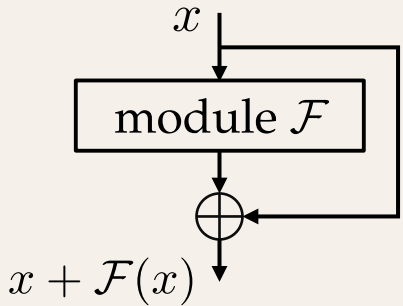
G can in *principle* be any group of transformations

Lift-Convolve-Project



CNNs share similarities with PDE solvers

Heat equation	Analytic solution	CNN
$\begin{cases} \frac{\partial f}{\partial t} = \nabla^2 f \\ f(0) = f_0 \end{cases}$	$f(t) = G_t * f_0$	$f_{out} = \sigma(K * f_{in})$

Additive residual connection	Forward Euler discretization
 $x + \mathcal{F}(x)$	$\dot{x}(t) = \mathcal{F}(x(t))$ is solved as $x(t + \Delta t) = x(t) + \Delta t \mathcal{F}(x(t))$

Idea

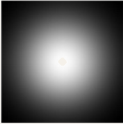
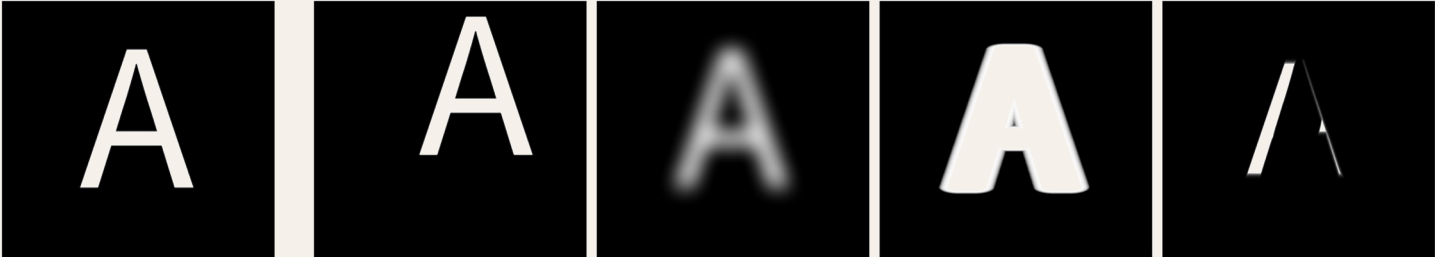
Take a PDE from the image processing world and put it in a neural network

$$\begin{cases} \frac{\partial u}{\partial t} = -\mathbf{c}u - (-\Delta_{\mathcal{G}_1})^\nu u + \frac{1}{\alpha} \|du\|_{\mathcal{G}_2^+}^\alpha - \frac{1}{\alpha} \|du\|_{\mathcal{G}_2^-}^\alpha \\ u(0) = u_0 \end{cases}$$

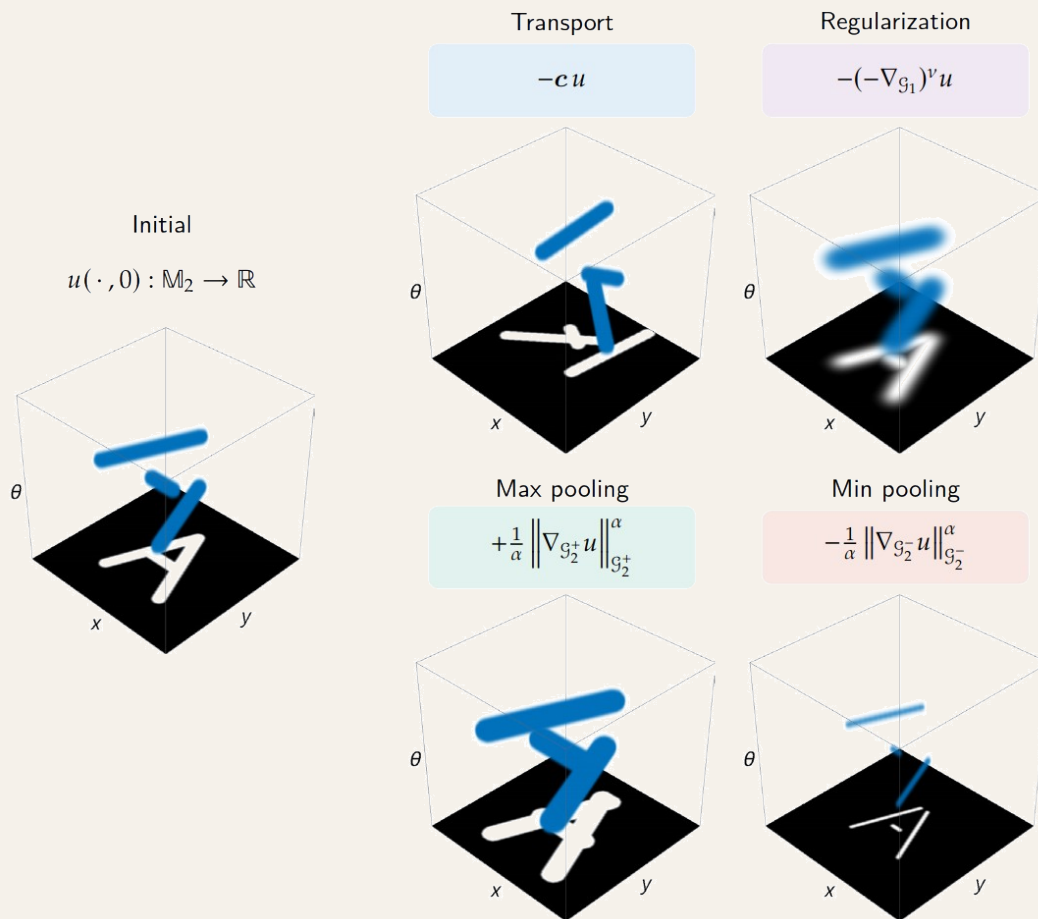
trainable: convection vector field $\mathbf{c}$ & metrics $\mathcal{G}_1, \mathcal{G}_2^+, \mathcal{G}_2^-$

hyper parameter $0 < \nu \leq 1, \alpha > 1$

Interpretation

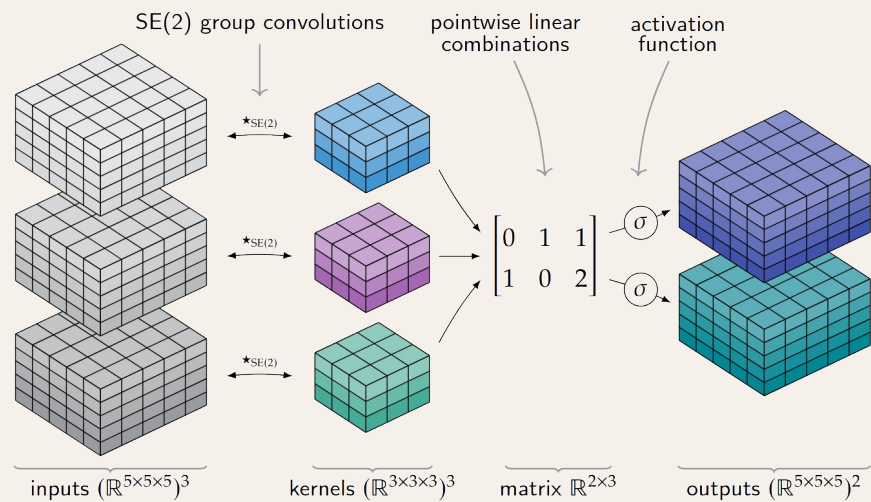
$\mathbb{R}^2$	Transport	Regularization	Max pooling	Min pooling
PDE Term:	$-cu$	$-(-\Delta_{g_1})^\nu u$	$+\frac{1}{\alpha} \ du\ _{g_2^+}^\alpha$	$-\frac{1}{\alpha} \ du\ _{g_2^-}^\alpha$
Parameters:	transport vector $\begin{pmatrix} 100 \\ 120 \end{pmatrix}$	Riemannian metric tensor $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	Riemannian metric tensor $\begin{pmatrix} 9 & 0 \\ 0 & 1 \end{pmatrix}$	Riemannian metric tensor $\begin{pmatrix} 5 & 11 \\ 11 & 40 \end{pmatrix}$
Operation:	resample with offset 	convolution with kernel 	dilation with kernel 	erosion with kernel 
				

The PDE generalizes to any homogeneous space

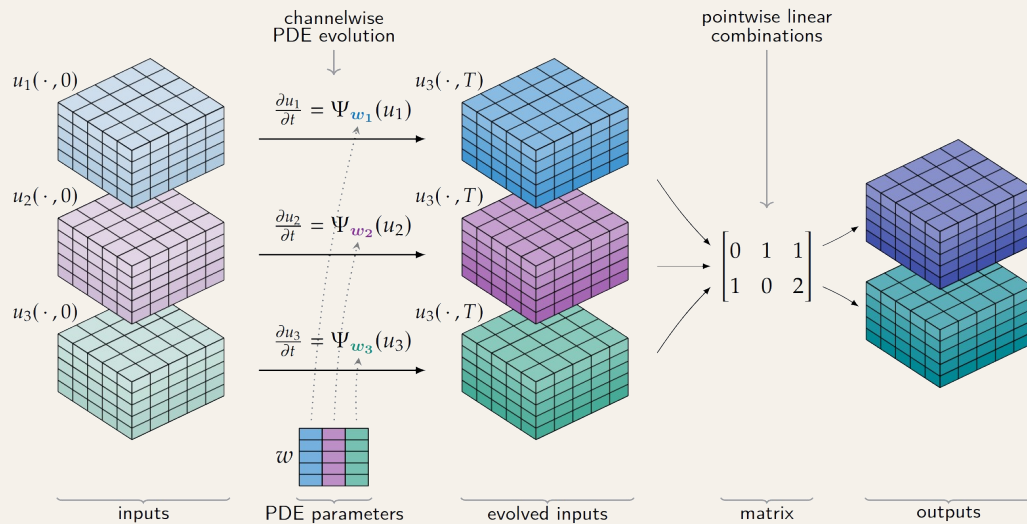


PDE-based layer

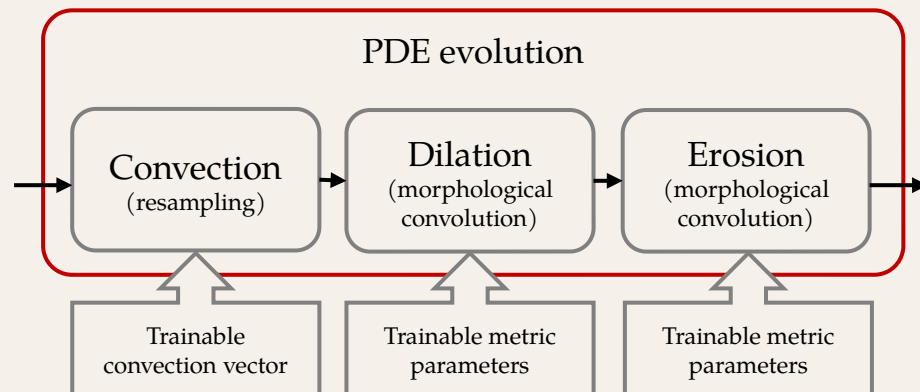
Depthwise CNN layer:



PDE-based layer:



PDE-based layer



Morphological convolution

On $\mathbb{R}^2$ for translation equivariance:

$$(k \boxplus f)(x) = \sup_{y \in \mathbb{R}^2} k(x - y) + f(y)$$

- dilation: $k \boxplus f$
- erosion: $-(-k \boxplus -f)$
- generalizes to the group setting

Morphological Group Convolution

DEFINITION 6.6 (Morphological convolution) Let G/H be a homogeneous space of a Lie group G , $f \in L^\infty(G/H)$ and let $k : G/H \rightarrow \mathbb{R} \cup \{\infty\}$ be proper (i.e. not everywhere ∞) then we define the *morphological convolution* of f with kernel k as:

$$(k \square_{G/H} f)(p) := \inf_{q \in G/H} \{(g_p \cdot k)(q) + f(q)\} \quad (6.29)$$

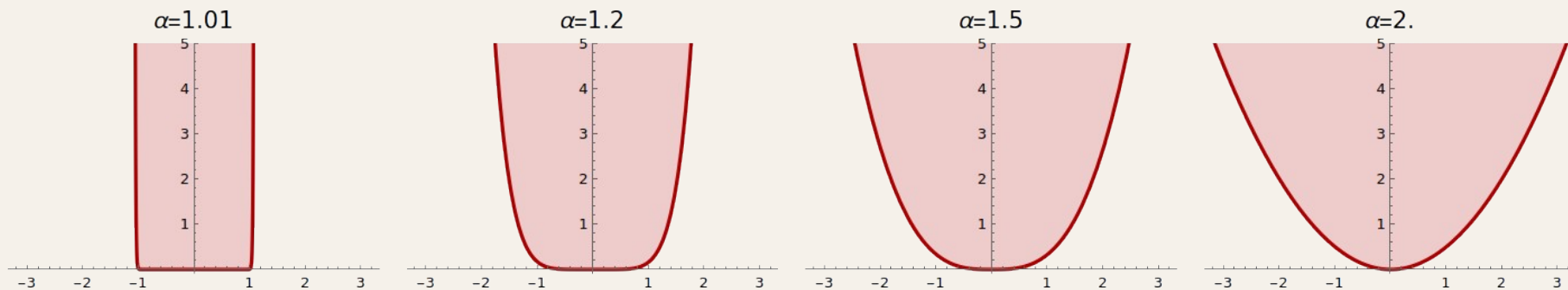
where $g_p \in p$ is freely chosen for all $p \in G/H$. For this expression to be well-defined we require that the kernel k satisfies the compatibility condition:

$$\forall h \in H : h \cdot k = k. \quad (6.30)$$

Morphological kernel

$$k_t^\alpha(p) = \frac{t}{\beta} \left(\frac{d_{\mathcal{G}}(p_0, p)}{t} \right)^\beta$$

- α, β controls sharpness $\frac{1}{\alpha} + \frac{1}{\beta} = 1$
- the metric contains the trainable parameters



Problem

Recomputing $p \mapsto d_{\mathcal{G}}(p_0, p)$ the whole time is expensive, instead:

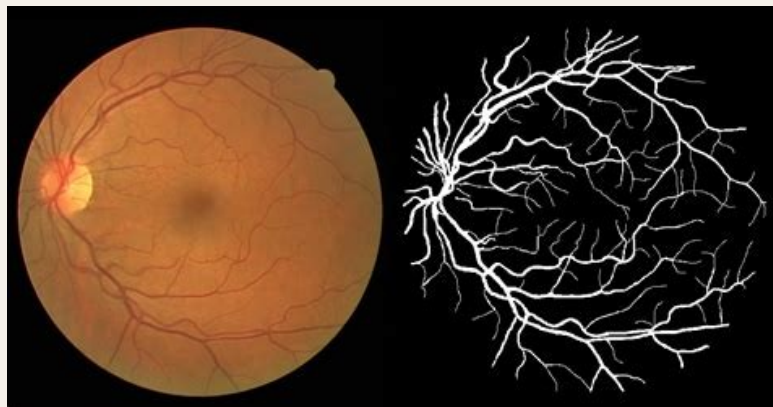
$$\rho_{\mathcal{G}}(p) := \inf_{g \in p} \|\log_G g\|_{\mathcal{G}} \approx d_{\mathcal{G}}(p_0, p)$$

where we interpret $p \in G/H$ and so $p \subset G$.

In the principal case this is just $\rho_{\mathcal{G}}(g) := \|\log_G g\|_{\mathcal{G}}$.

Essentially: estimate distance by the shortest exponential curve.

Retinal vessel segmentation



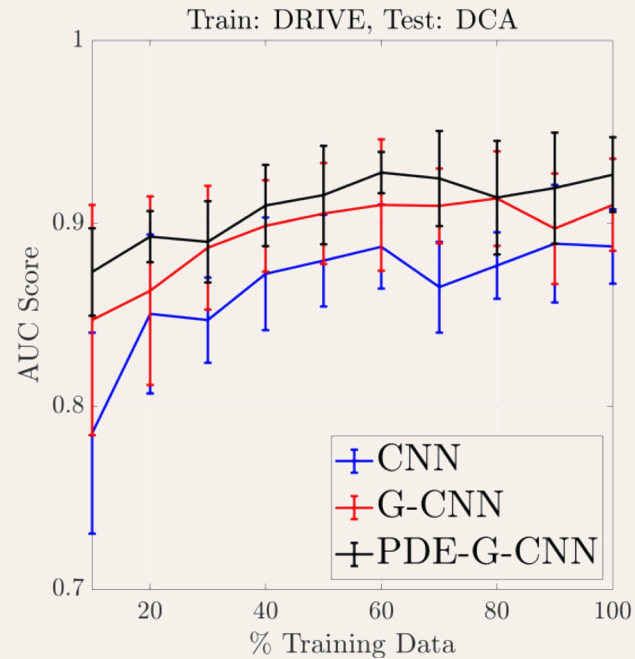
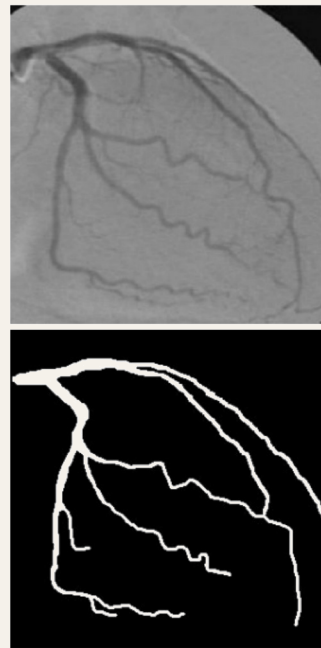
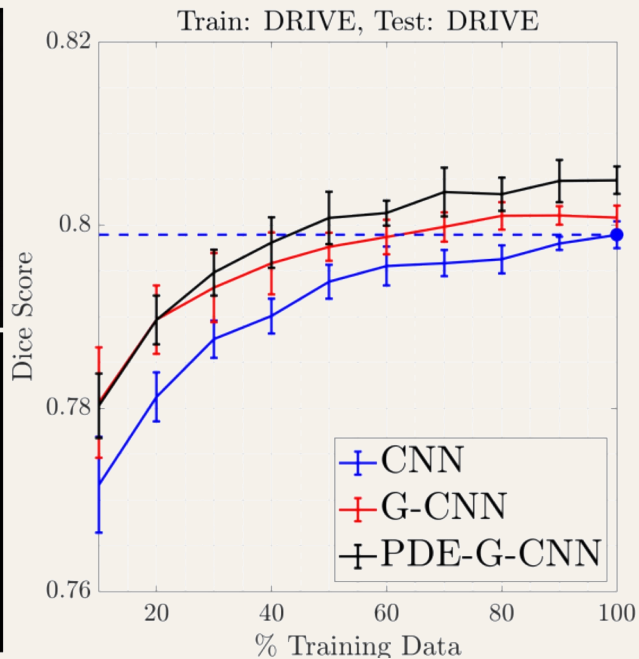
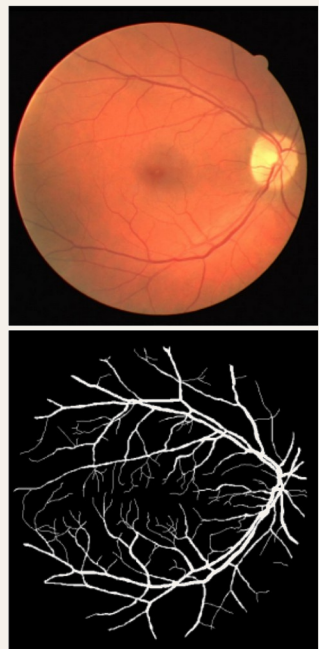
Model (transformation group)	Parameters	DICE score %
CNN 6 T(2)	47,352	80.58 $\pm$ 0.17
G-CNN 6 SE(2)	39,258	80.85 $\pm$ 0.22
CDE-PDE-CNN 6 SE(2)	4,128	81.15 $\pm$ 0.18
CNN 12 T(2)	129,432	81.89 $\pm$ 0.05
G-CNN 12 SE(2)	114,378	81.92 $\pm$ 0.12
CDE-PDE-CNN 12 SE(2)	3,768	82.20 $\pm$ 0.07

RotNIST digit classification



Model (transformation group)	Parameters	Accuracy %
LeNet 5 T(2)	44,426	97.42 $\pm$ 0.66
G-CNN Classifier 4 SE(2)	12,700	98.86 $\pm$ 0.21
CDE-PDE-CNN Classifier 4 SE(2)	2,542	98.90 $\pm$ 0.18
CDE-PDE-CNN Classifier 4 T(2)	2,518	98.96 $\pm$ 0.07

Limited training data & generalization



Strengths & weaknesses

- + fewer parameters
- + geometric interpretability
- + need less data
- + equivariance
- computational complexity / memory capacity in higher dimensions
- strong geometric prior can be counter productive

PyTorch extension



- > `conda install lietorch::lietorch`
- > `pip install lietorch`
- > `docker pull ghcr.io/bmnsnets/lietorch-ssh`



gitlab.com/bsmetsjr/lietorch

```

import torch.nn as nn
import lietorch.nn as lnn

C = 8
n_classes = 2
kernel_size = (5, 5, 5)
alpha = 0.65

model = nn.Sequential(
    # lift 2D image [B,3,H+3,W+3] to the space of positions and orientations [B,C,8,H,W]
    lnn.LiftM2Cartesian(in_channels=3, out_channels=C, orientations=8, kernel_size=7),

    # PDE layers in M2
    lnn.CDEPdeLayerM2(C, C, kernel_size, alpha),
    lnn.CDEPdeLayerM2(C, C, kernel_size, alpha),

    # project M2 feature map [B,C,8,H,W] back to 2D [B,C,H,W]
    lnn.MaxProjectM2(),

    # make pointwise linear combinations to class logits [B,n_classes,H,W]
    nn.Conv2d(C, n_classes, 1, 1, bias=False),
)

# a CDEPdeLayerM2 module is equivalent to:
nn.Sequential(
    lnn.ConvectionM2(C),
    lnn.FractionalDilationM2(C, kernel_size, alpha),
    lnn.FractionalErosionM2(C, kernel_size, alpha),
    lnn.LinearM2(C, C), # pointwise linear combination
    nn.BatchNorm3d(C), # optional
)

```

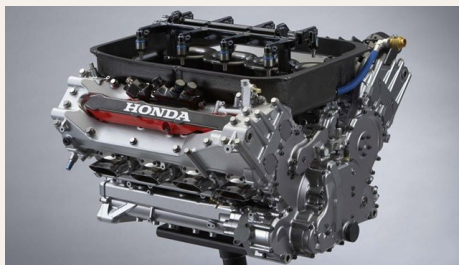
Mathematics & AI : The Big Picture



Theory of matter



Thermodynamics



 deepseek
R1

Theory of
Deep Learning?



Thank you