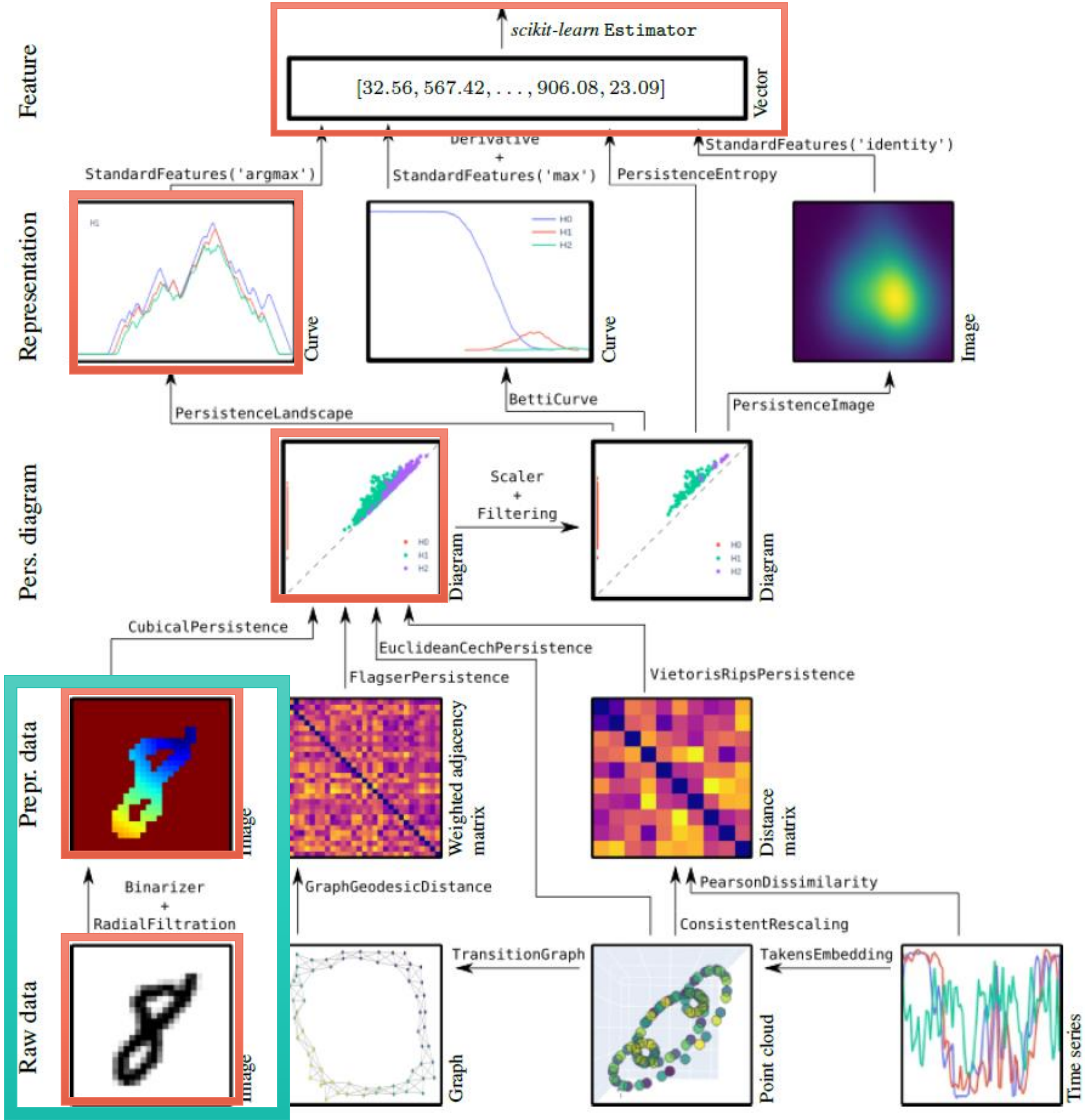
A CONVOLUTIONAL PERSISTENCE TRANSFORM

Elchanan Solomon
Duke University, informal Barcelona Campus
joint work with Paul Bendich

The TDA Pipeline (part of)

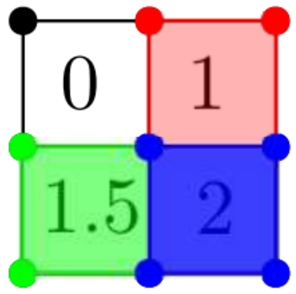


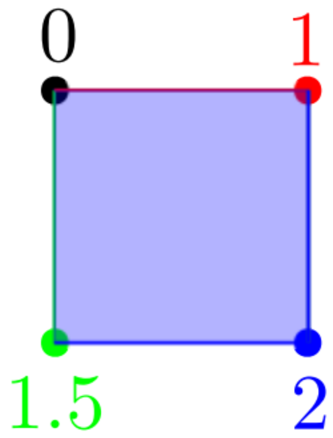
- Describe what this step usually consists of.
- Explain why the standard approach is limited.
- Propose a new approach.
- Prove that new approach is better.
- Try the new approach on real data and see the results.

https://docs-tda.giotto.ai/0.5.0/library.html

Converting Images to Filtered Simplicial Complexes

- Represent the image grid using a cubical complex.
- Pixels are either vertices or top-dimensional cells. Pixel intensity defines a function on vertices/top-dim cells.
- Extend to the rest of the complex using lower/upper-star.
- Use Freudenthal triangulation .





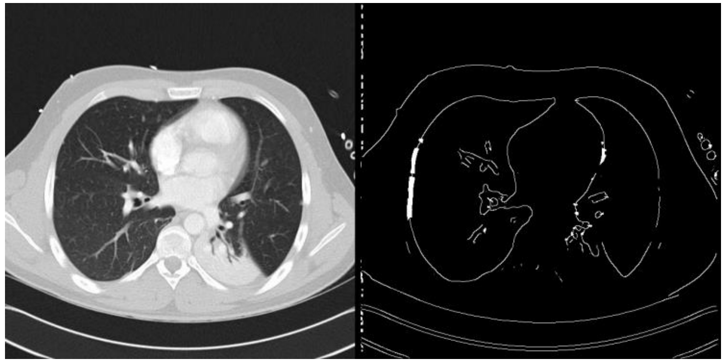
The two constructions give different but equivalent diagrams.

The Persistent Homology of Dual Digital Image Constructions

Bea Bleile[1], Adélie Garin[2], Teresa Heiss[3], Kelly Maggs[2], Vanessa Robins[4]

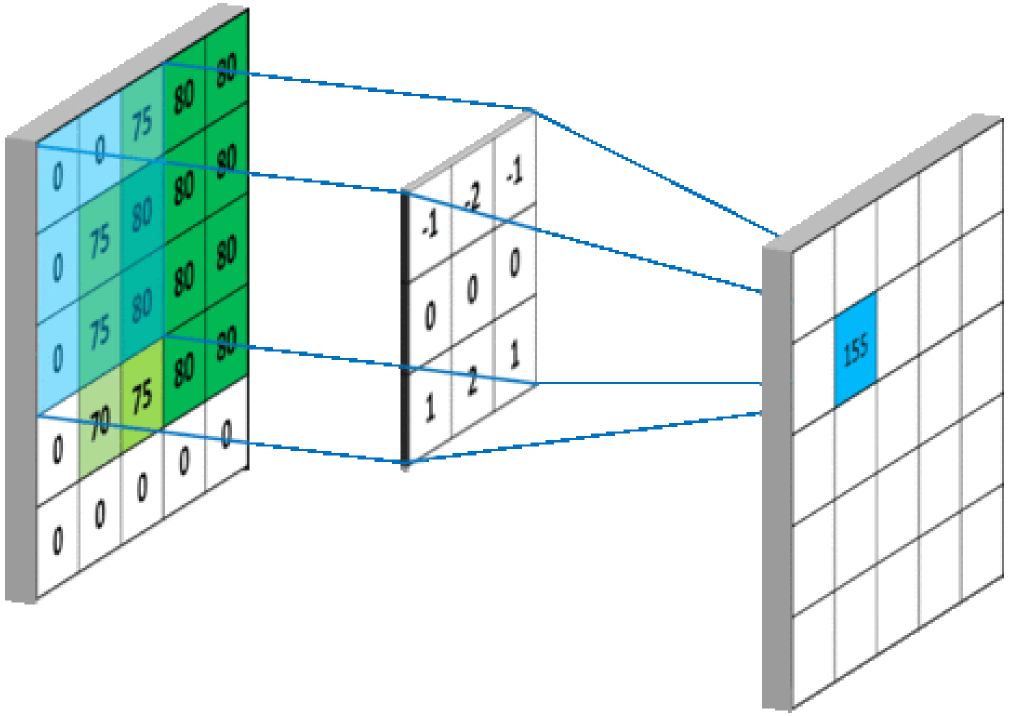
Limitations of Image TDA (Persistence)

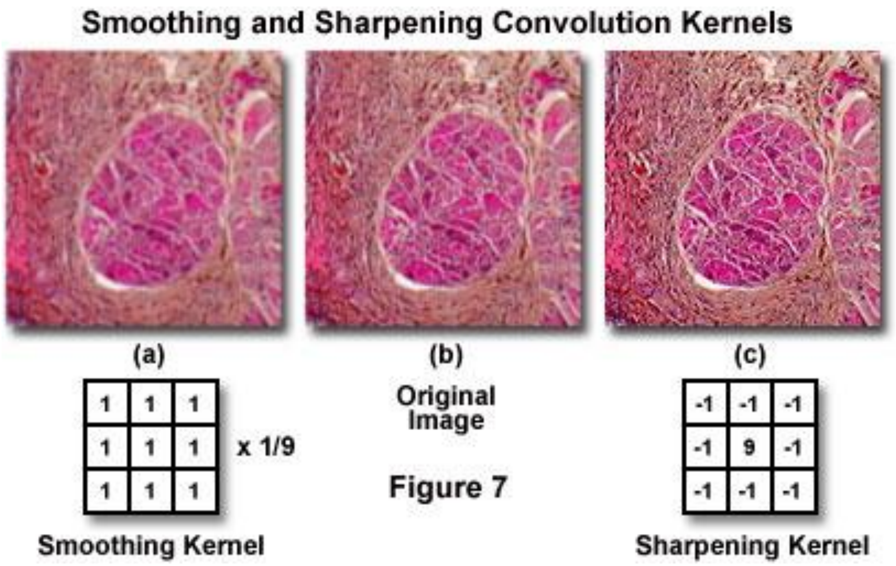
- Persistence is ***unstable to outliers***.
- Many different images can have the same persistence. In other words, image persistence is ***lossy / not injective***.
- Image persistence is ***not customizable***. You get a single invariant for each image, and you cannot adapt that invariant depending on the context.
- Some images need ***more preprocessing*** for topology to be useful.



https://epochabuse.com/canny-edge-detection/

Adding an Intermediate Step: Preprocess with Convolutions





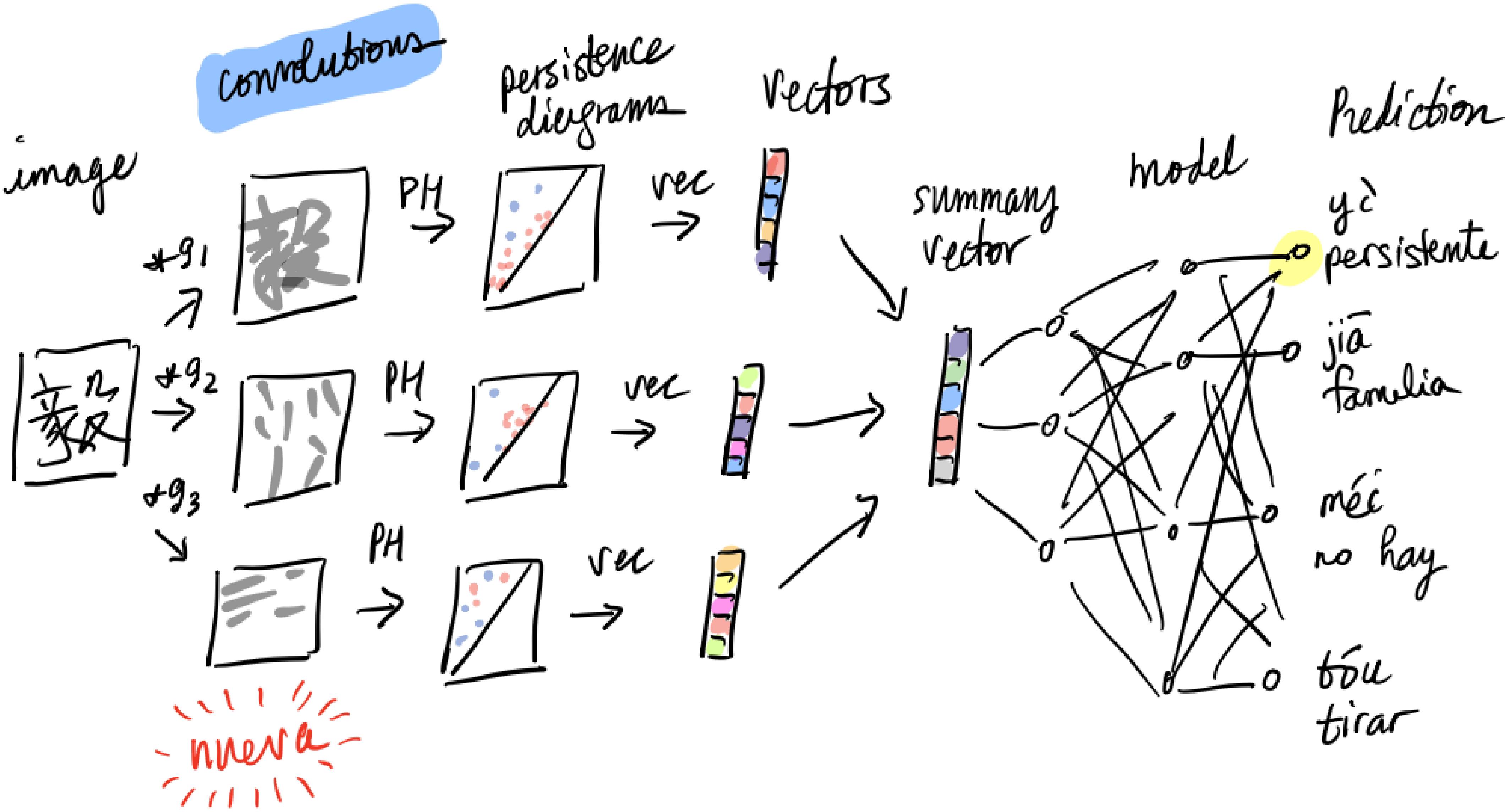
Convolutions + Persistence

=

Convolutional Persistence

https://www.olympus-lifescience.com/zh/microscope-resource/primer/digitalimaging/imageprocessingintro/

https://www.cs.cmu.edu/~bhiksha/courses/deeplearning/Spring.2019/archive-f19/www-bak11-22-2019/document/note/hwnotes/hw2/hw2.html


convolutions
persistence diagrams
Vectors
image
*g1
*g2
*g3
PH
PH
PH
vec
vec
vec
summary vector
model
Prediction
yì persistente
jiā familia
méi no hay
tóu tirar
nueva

Properties of Convolutional Persistence

- Using the right filters, convolutional persistence is ***more stable to outliers.***
- A convolved image is often much smaller than the original. This greatly ***speeds up*** persistence computations.

A Fast and Robust Method for Global Topological Functional Optimization

Elchanan Solomon*
Department of Mathematics,
Duke University
Durham, USA
yitzchak.solomon@duke.edu

Alexander Wagner*
Department of Mathematics,
Duke University
Durham, USA
alexander.wagner@duke.edu

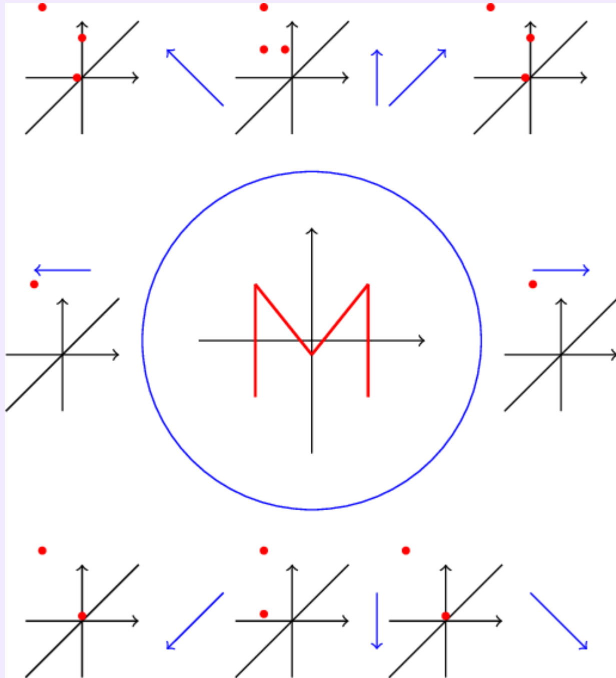
Paul Bendich
Department of Mathematics, Duke University
Geometric Data Analytics
Durham, USA
paul.bendich@duke.edu

- For any two images, there exists some filter that can tell them apart – meaning that convolving with that filter produces different persistence diagrams. (***injectivity***)

The Proof of Injectivity

We use another construction from applied topology: the persistent homology transform.

For a vector $v \in \mathbb{S}^{d-1} \subset \mathbb{R}^d$, define the function $f_v = \langle \cdot, v \rangle$.
Given $X \subset \mathbb{R}^d$, define $\mathrm{PHT}(S)$ be the function sending vectors $v \in \mathbb{S}^{d-1}$ to their corresponding persistence diagrams $PH(X, f_v)$.

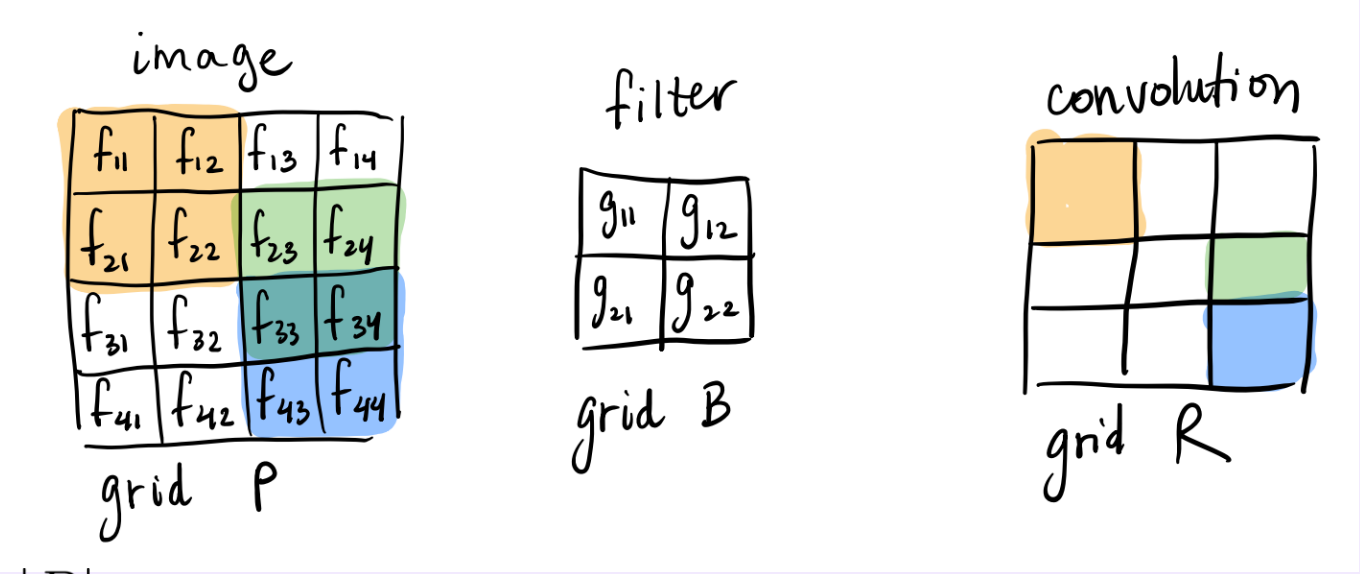


Theorem (Turner,Mukherjee,Boyer,Ghrist,Levanger,Mai,Curry):

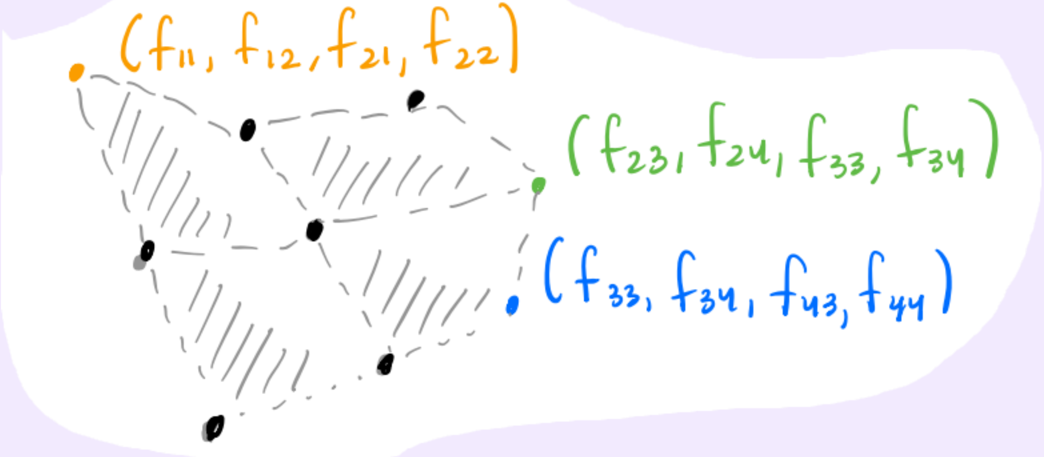
The PHT is injective, i.e. $PHT(X) = PHT(Y)$ implies $X = Y$.

https://ieeexplore.ieee.org/document/8205008

Relating Convolutions and the PHT

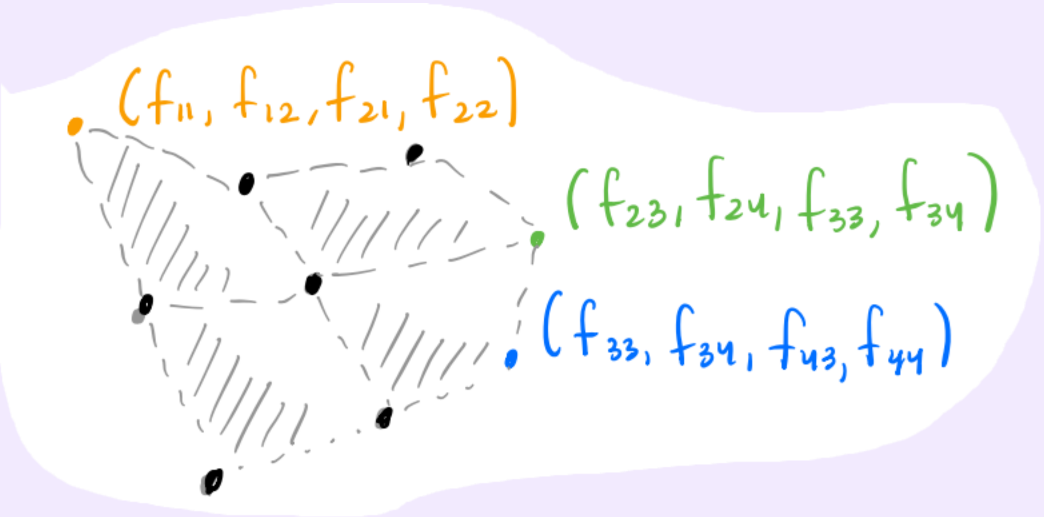


Define $\iota_f : R \to \mathbb{R}^{|B|}$:



Define $\vec{g} = (g_{11}, g_{12}, g_{21}, g_{22})$.
Observe: For a vertex $r \in R$, we have $(f * g)(r) = \iota_f(r) \cdot \hat{g}$.

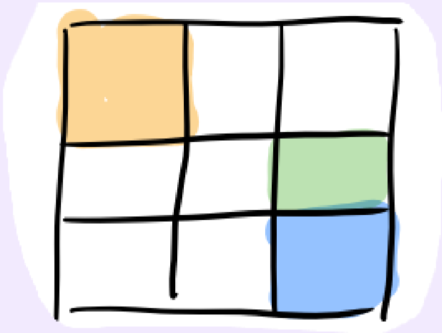
The continuously embedded grid.



$$\cong$$

(generically)

The discrete, abstract grid.



Continuous filtration $\langle \cdot, \vec{g} \rangle$.

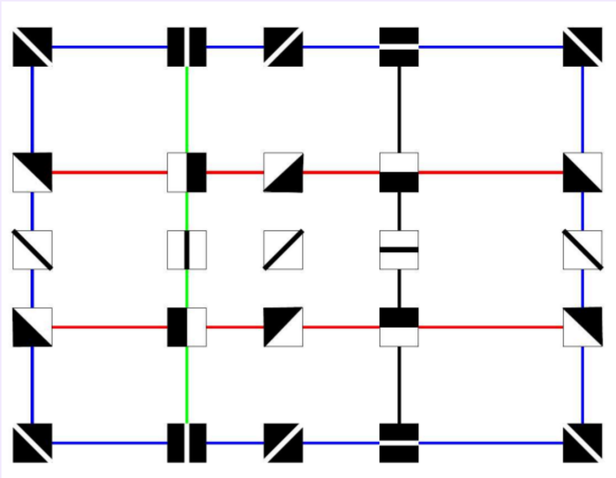
Discrete filtration $f * g$.

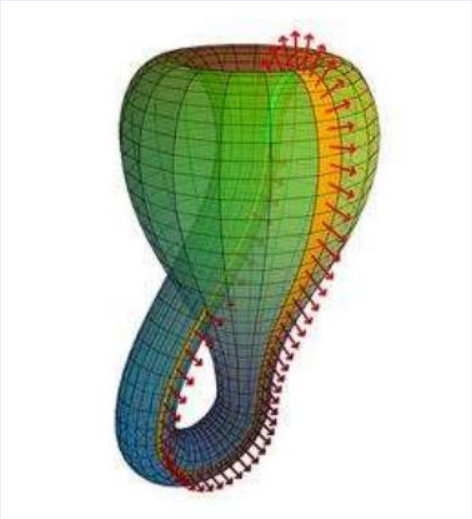
Due to a well-known result in persistence theory, these two filtrations, which agree on the vertex set, produce identical persistence diagrams.

$$PHT(\iota_f(R)) \text{ evaluated at } \vec{g}. \quad = \text{Convolutional persistence for filter } g.$$

Using the embedding ι_f, convolutional persistence can be viewed as a special case of the PHT, and so inherits its injectivity properties. This can easily be generalized to high-dimensional images with multiple channels.

Important Observation: No extra information is gained by using filters that are orthogonal to the patches found in the images. In natural images, the space of patches has large codimension, significantly reducing the number of filters needed.





https://link.springer.com/article/10.1007/s11263-007-0056-x

Convolutional Persistence Pipeline

- Given: collection of images.
- Pick: family of n filters $\mathcal{G}$. Which filters?
- For each image, convolve with $\mathcal{G}$ to get n new images, and then compute persistence, obtaining n persistence diagrams.
- Vectorize the collection of diagrams. How to vectorize?
- Use vectors in Machine Learning model. Which model?

Pipeline Parameters

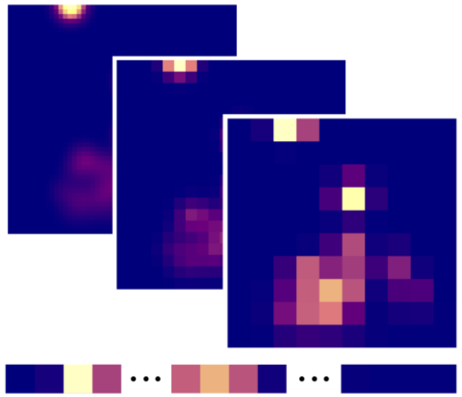
Filters

1. [1] – trivial filter.
2. Standard image processing filters.
3. Eigenfilters: Apply PCA to space of image patches and take random linear combinations of top eigenvectors.
4. Random filters.

Vectorizations

1. Persistence Images.
2. Total persistence.

A. Concatenating vectors
B. Averaging vectors



https://www.math.colostate.edu/~adams/research/

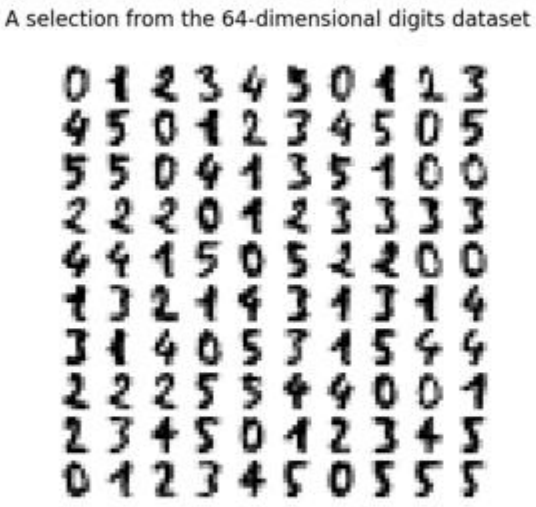
Models

1. k-NN.
2. Boosted Tree.
3. Neural Network.

No feature engineering, dimensionality reduction, model tuning, etc.

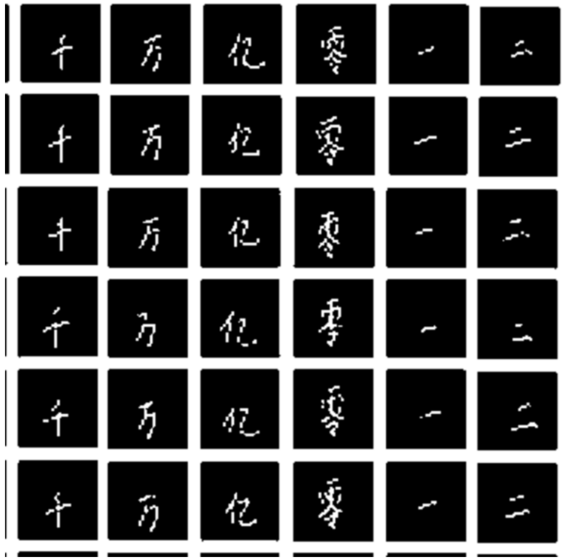
Classification Tasks

UCI digits

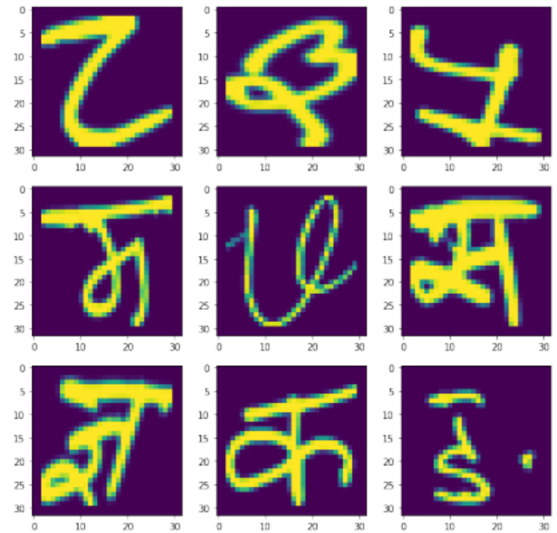


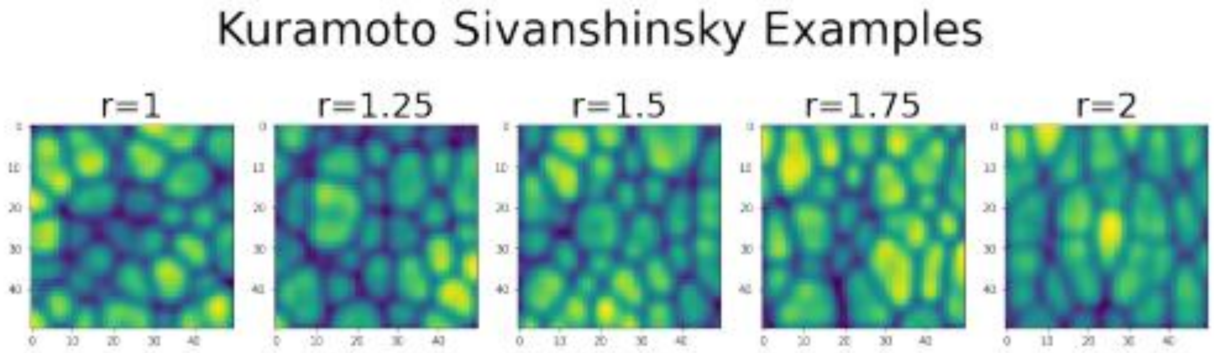
MNIST

Chinese Digits



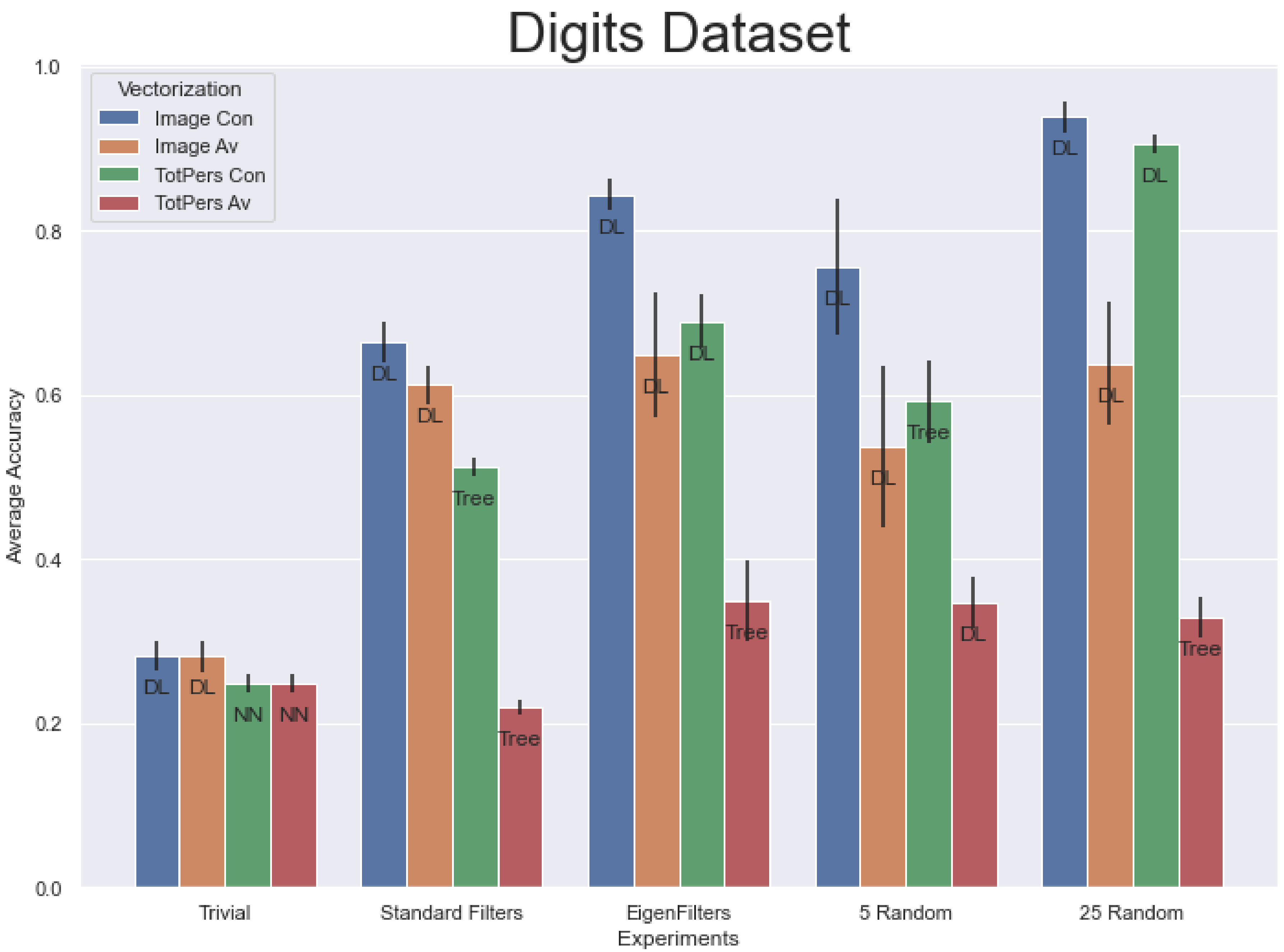
Devanagari “MNIST”



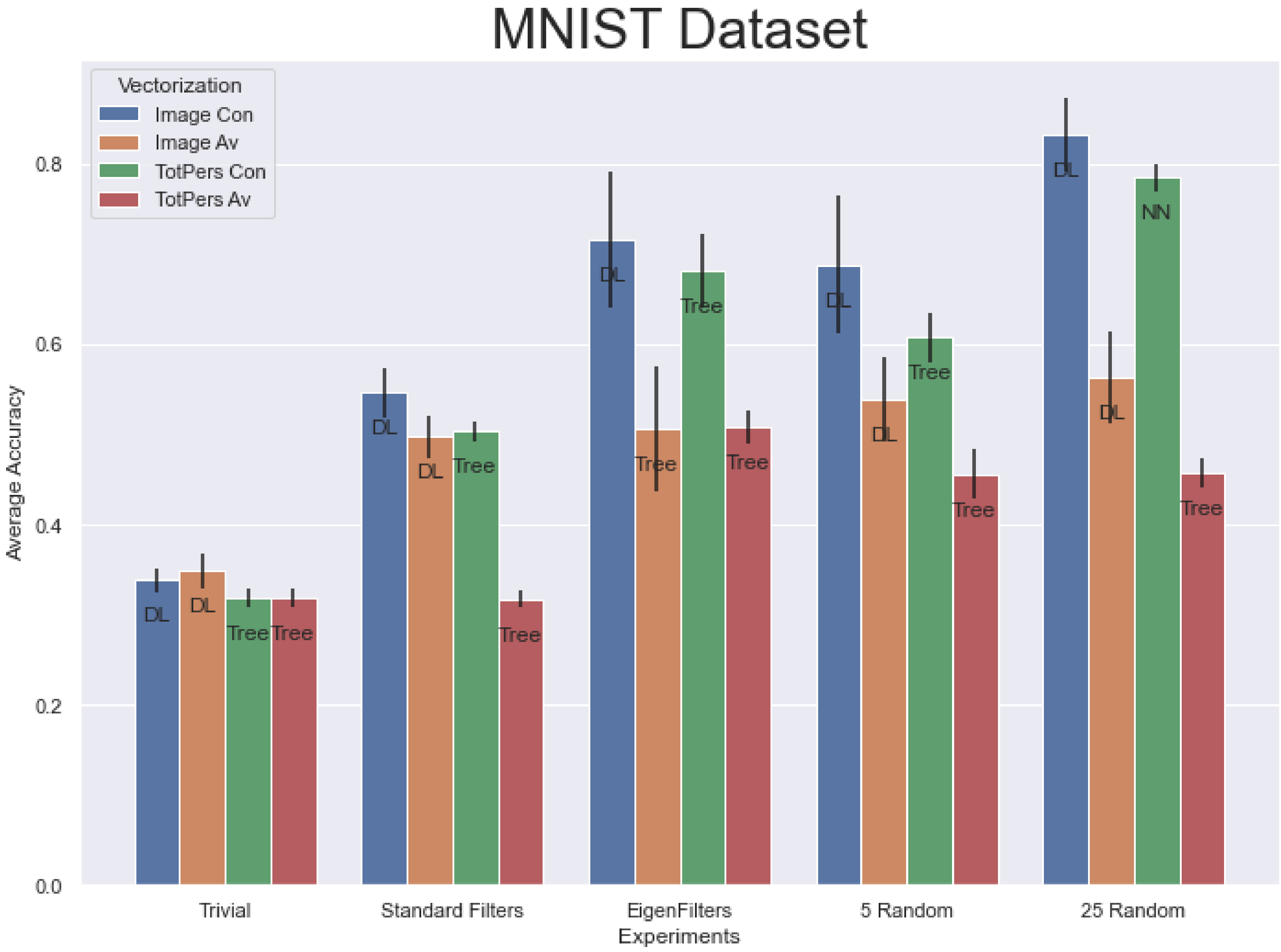


$$u_t = -\nabla^2 u - \nabla^2 \nabla^2 u + r(u_x)^2 + (u_y)^2$$

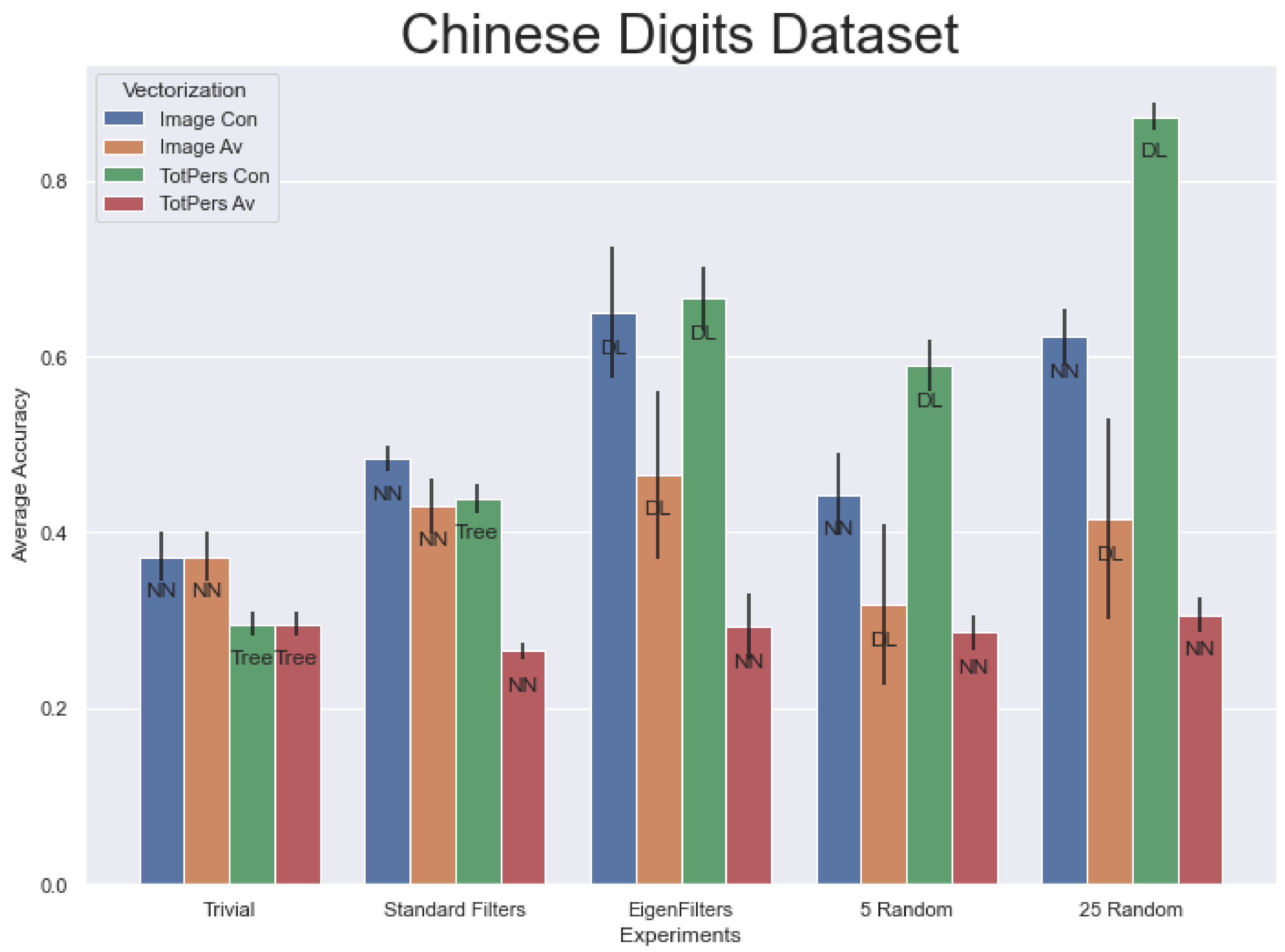
https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits
http://yann.lecun.com/exdb/mnist/
https://www.kaggle.com/datasets/gpreda/chinese-mnist
https://www.kaggle.com/datasets/anurags397/hindi-mnist-data



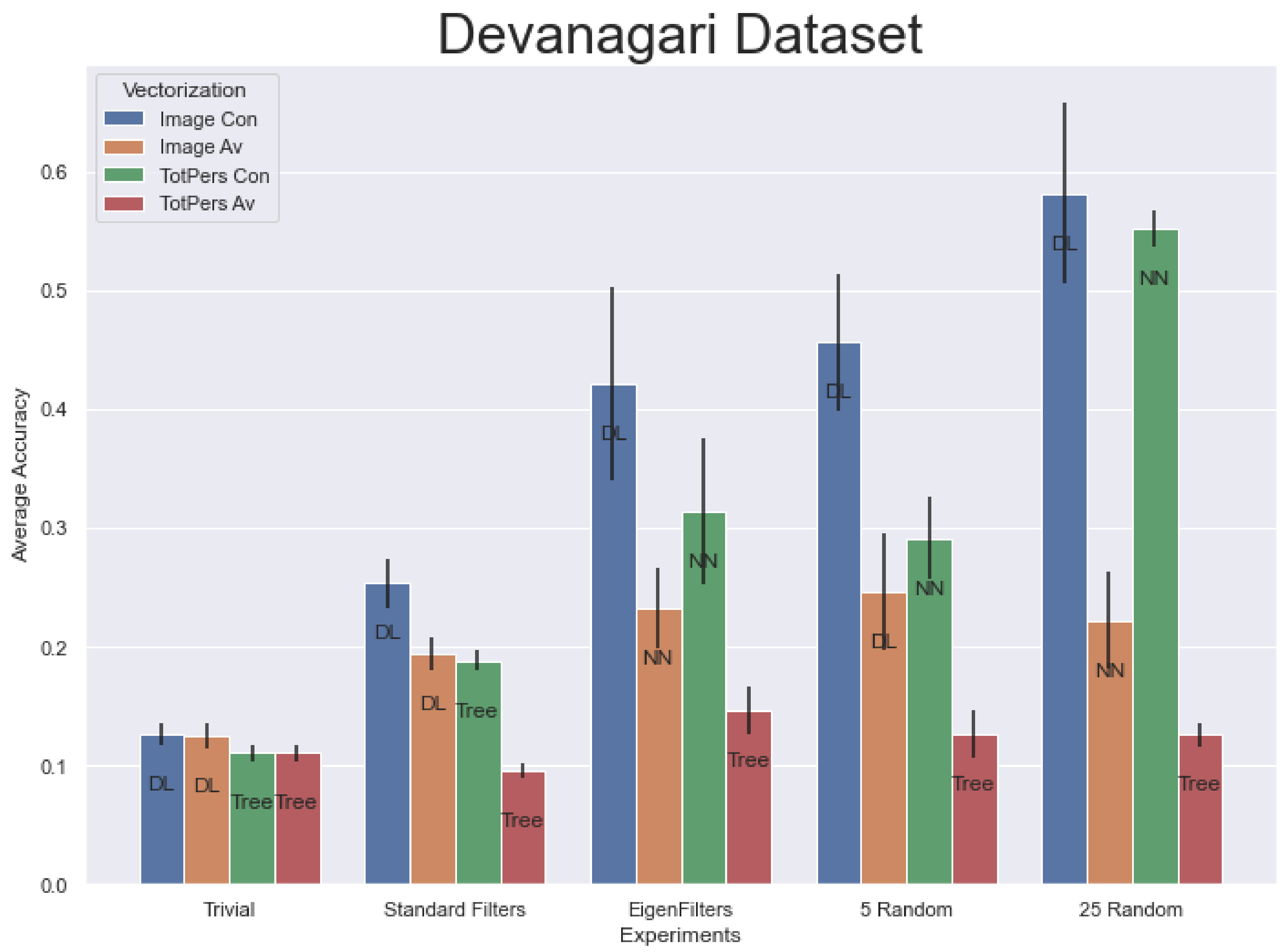
Digits Dataset
Vectorization
Image Con
Image Av
TotPers Con
TotPers Av
Average Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
DL
DL
NN
NN
DL
DL
Tree
Tree
DL
DL
DL
Tree
DL
DL
Tree
DL
DL
DL
DL
Tree
Trivial
Standard Filters
EigenFilters
5 Random
25 Random
Experiments



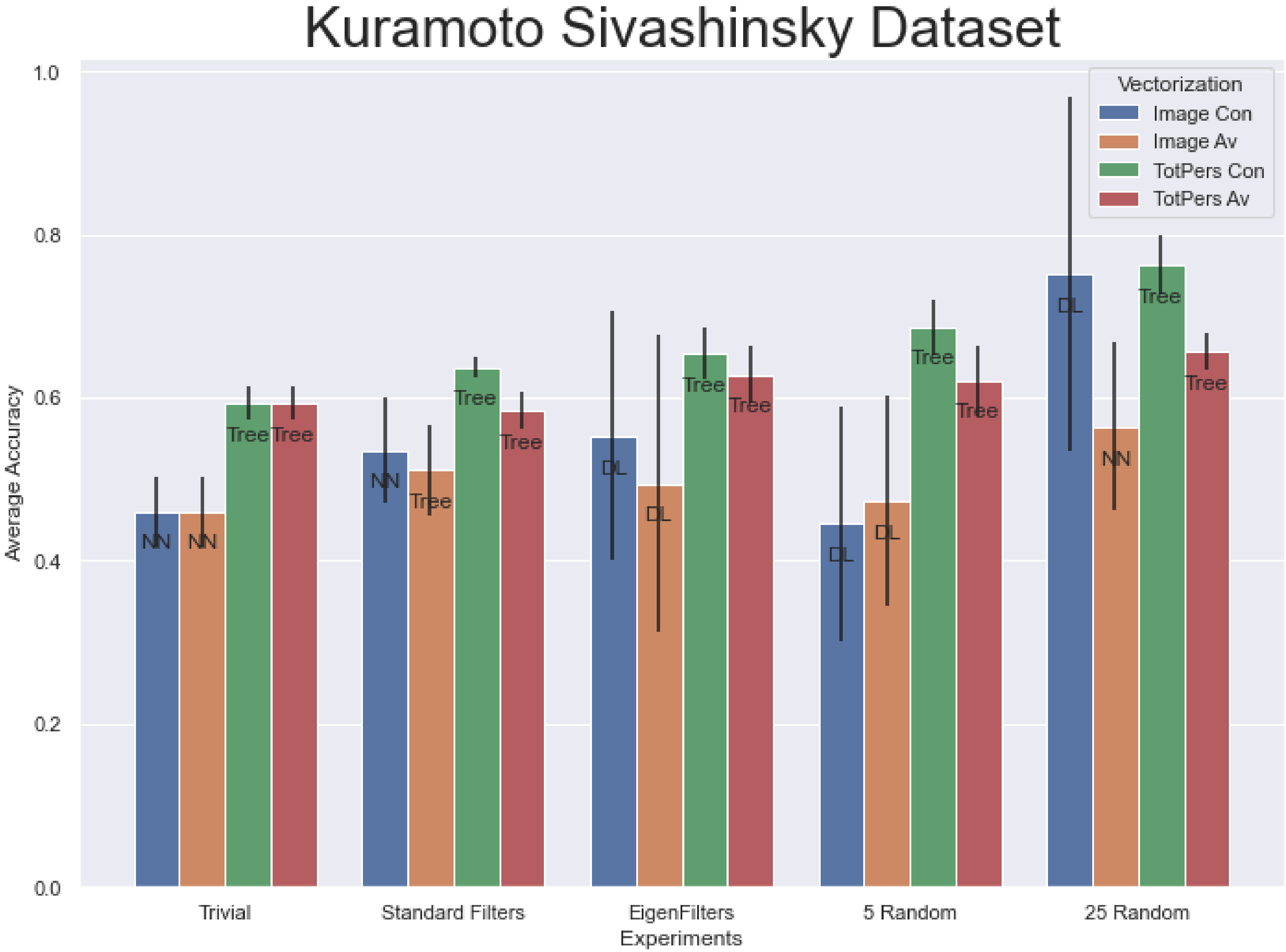
MNIST Dataset
Vectorization
Image Con
Image Av
TotPers Con
TotPers Av
Average Accuracy
0.0
0.2
0.4
0.6
0.8
DL
DL
Tree
Tree
DL
DL
Tree
Tree
DL
Tree
Tree
Tree
DL
DL
Tree
Tree
DL
DL
NN
Tree
Trivial
Standard Filters
EigenFilters
5 Random
25 Random
Experiments



Chinese Digits Dataset
Vectorization
Image Con
Image Av
TotPers Con
TotPers Av
Average Accuracy
0.0
0.2
0.4
0.6
0.8
NN
NN
Tree
Tree
NN
NN
Tree
NN
DL
DL
DL
NN
NN
DL
DL
NN
NN
DL
DL
NN
Trivial
Standard Filters
EigenFilters
5 Random
25 Random
Experiments



Devanagari Dataset
Vectorization
Image Con
Image Av
TotPers Con
TotPers Av
Average Accuracy
0.0
0.1
0.2
0.3
0.4
0.5
0.6
DL
DL
Tree
Tree
DL
DL
Tree
Tree
DL
NN
NN
Tree
DL
DL
NN
Tree
DL
NN
NN
Tree
Trivial
Standard Filters
EigenFilters
5 Random
25 Random
Experiments



Kuramoto Sivashinsky Dataset
Vectorization
Image Con
Image Av
TotPers Con
TotPers Av
Average Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
NN
NN
Tree
Tree
NN
Tree
Tree
Tree
DL
DL
Tree
Tree
DL
DL
Tree
Tree
DL
NN
Tree
Tree
Trivial
Standard Filters
EigenFilters
5 Random
25 Random
Experiments

Observations

- Convolution persistence significantly outperforms ordinary image persistence. Concatenating vectors and using deep learning seems to give the best accuracy.
- Eigenfilters are best, but random filters also work really well.

Why?

- Total persistence is a very effective vectorization.

Why??

More questions…

MOLTES GRÀCIES

- How does the filter size affect the results above?
- What happens if you use other vectorizations: Euler curves, persistence landscapes, etc. or allow learnable vectorizations?
- Can you get better accuracy scores by incorporating feature engineering and model tuning?
- Can you learn optimal filters using the training data?
- Run these experiments on more complex, higher-dimensional, or multi-channel data where topology is already known to be useful.
- There are many technical results and open questions surrounding the PHT. Can these be improved/answered for convolutional persistence?

https://arxiv.org/abs/2208.02107
https://github.com/yesolomon/convpers