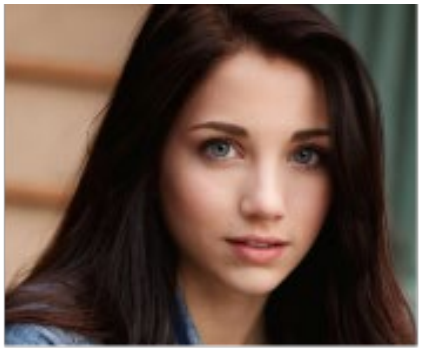
Rotation-Invariant CNN using scattering transform for image classification

Authors: Rosemberg Rodriguez, Petr Dokladal, Eva Dokladalova

Université Paris-Est, ESIEE Paris, LIGM UMR CNRS 8049

PSL Research University—MINES ParisTech

Naturally un-oriented data



People Faces

Plankton[1]

Food[2]

Normally up-right

Naturally un-oriented data

- While some data is naturally upright oriented (faces or numbers) other presents random orientations (plankton, galaxies, food).
- When invariance to symmetries (rotations, change of scale, symmetry) is required the models become more complicated.

[1]WHOI-Plankton - https://beagle.whoi.edu/redmine/projects/ifcb-man/wiki/WHOI-Plankton
[2]FoodAI - http://images.nvidia.com/content/APAC/events/ai-conference/resource/ai-for-research/FoodAI-Food-Image-Recognition-with-Deep-Learning.pdf

Outline

- State of the art
- Expected output results
- Overall method
    1. Scattering transform oriented features
    2. Vectorization
    3. Convolutional trained predictors
    4. Probability matrix caracteristics
- Study cases
    1. Naturally random oriented data
    2. Normally up-right data
- Conclusions

State of the art

The existing approaches to encode rotation equivariance into CNNs can be divided in two families:

1. Transform the representation on input image or feature maps.

2. Rotating the internal filters.

Transform the representation on input image

These methods have the advantages of exploiting conventional CNN implementations, one of the most popular approach is contained here: data augmentation [1].

The main limitations of these methods:

- The algorithm still need to learn feature representation separately for different variations of the original data. E.g. Edge-detecting features[2] under rotation-invariance setting still need to learn separately vertical and horizontal edges.

- Some transformations of the data can actually result in the algorithm learning from noise samples or wrong labels. E.g. random crops applied to the input can capture a non-interest region.

- The more variations are considered in the data, the more flexible the model needs to be to capture all the variations in the data.

[1] **D.A. Van Dyk and X.L. Meng.** *The art of data augmentation.* Journal of Computational and Graphical Statistics, 10(1), 2001.
[2] **D.Ciresan, A. Giusti L.M. Gambardella and J. Schmidhuber.** *Deep neural networks segment neuroanl membranes in electron microcopy images..* Advances in neural information processing systems.

Rotating the filters

Multiple existing approaches seek to encode rotational equivariance into CNNs. Many of these follow a broad approach of introducing filter or feature map copies at different rotations.

Some approaches included are:

- **Steerable filters :**

  "An efficient architecture that synthetizes filters of arbitrary orientations from linear combinations of basis filters, allowing one to adaptatively "steer" a filter to any orientation."[1]

  Harmonic Networks [2] capture explicitly the underlying orientations with complex circular harmonics, finding the orientation by the conjunction of multiple discrete filters.

- **Encode equivariance to discrete rotations:**

  These approaches try to find a compromise between computational resources required and the amount of information kept by the layers, by keeping the model shallow or accounting for a limited amount of orientations.

  Other approaches include feeding in multiple copies of the CNN input and merge the output predictions or trying to solve the problem by copying each feature map at four 90° rotations[3].

[1]**Freeman and Adelson.** *The design and use of steerable filters.* IEEE on Pattern Analysis and Machine Intelligence 1991
[2]**Worral, Garbin, Turmukhambetoc, Brostow.** *Harmonic Networks: Deep Translation and Rotation Equivariance.* CVPR 2017
[3]**B. Fasel and D. Gatica-Perez,** *Rotation invariant neoperceptron.* ICPR 2016, Hong Kong, China.

State of the art

Method	Error rate (in %)
SVM[1]	10.38±0.27
Harmonic Networks[2]	1.69
TI-Pooling[3]	1.2
Rotation equivariant vector field networks[4]	1.09
Oriented Response Networks[5]	0.76
Rotationally Invariant Conv. Module[6]	3.51

While the results of this approaches are good, all of them are trained on the commonly used variation of MNIST that is used for validating rotation-invariant algorithms, MNIST-rot.

This dataset contains 12000 training images and 50000 test samples.

All the samples are **rotated by a random angle from 0 to 2π.**

[1] **H. Larochelle et al.** *An empirical evaluation of deep architectures on problems with many factors of variation.* ICML 2007 pages 473 - 480
[2] **Worral, Garbin, Turmukhambetoc, Brostow.** *Harmonic Networks: Deep Translation and Rotation Equivariance.* CVPR 2017
[3] **D. Laptev, N. Savinov, J. M. Buhman** *TI-Pooling: transformation-invariant pooling for feature learning in convolutional neural networks.* CVPR 2016 Pages 289 - 297.
[4] **D. Marcos, M. Volpi, N. Komodakis, D. Tuia** *Rotation equivariant vector field networks.* arXiv: 1612.09346v3.
[5] **Y. Zhou et al.** *Oriented Responde Networks* CCVPR 2017 Pqges 4961 -4970
[6] **P. Follman and T. Bottger** *A rotationally-invariant convolution module by feature map backrotation* WACV 2018 pqges 784 - 792

State of the art

Method	Error rate (in %)
ORN-8(ORPooling)[1]	16.67
ORN-8(ORAlign)[1]	16.24
RotInv Conv. (RP_RF_1)[2]	19.85
RotInv Conv. (RP_RF_1_32)[2]	12.20

Results when trained only with upright oriented samples and validated on randomly rotated samples. The accuracy is lower.

Some of these methods use more than 1 million of trainable parameters and can not reach higher accuracy.

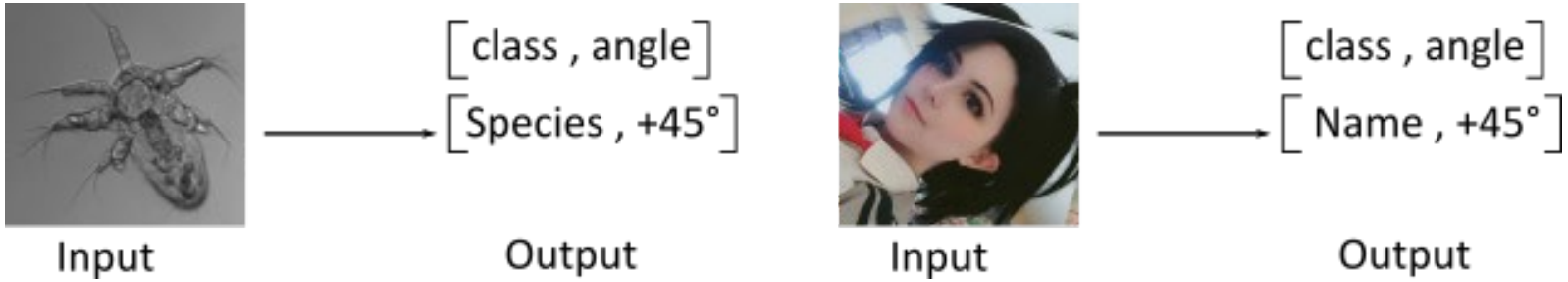
[1] **Y. Zhou et al.** *Oriented Responde Networks* CCVPR 2017 Pqges 4961 -4970
[2] **P. Follman and T. Bottger** *A rotationally-invariant convolution module by feature map backrotation* WACV 2018 pqges 784 - 792

Expected results

The main challenge of this work is to design a deep learning network architecture capable of :

1) Classifying rotated data with unlabeled orientation.

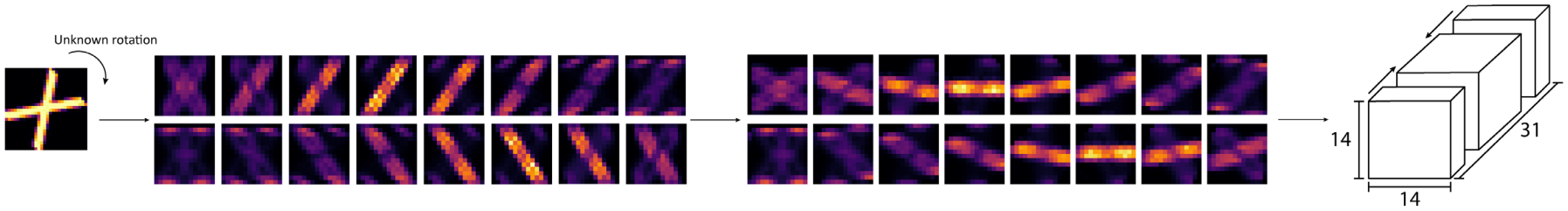
2) Training from a database with all “up-right” images and testing on randomly rotated samples.



Our proposal will use the roto-translational output space properties of the scattering transform.

The network should learn the rotation invariance without the use of data augmentation techniques.

Deep neural network proposal



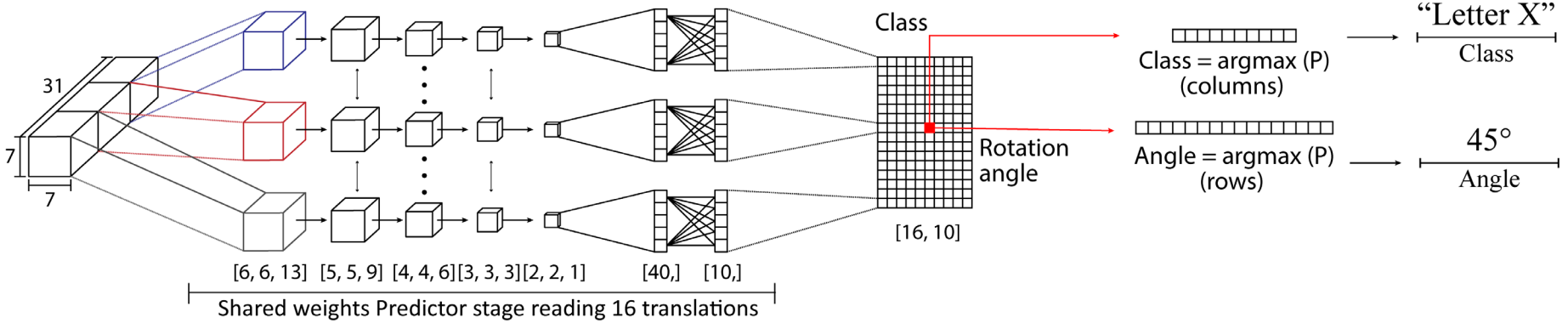
(a) Input pattern

(b) Oriented wavelet feature space
[14, 14, 16]

(c) Horizontally-oriented wavelets
[14, 14, 16]

(d) Periodic padding
[14, 14, 31]

Roto - translation covariance



(e) MaxPooling3D
[7, 7, 31]

(f) (5x) 3D Convolution
[2, 2, 1, 10]

(g) Dense Layer
[10,]

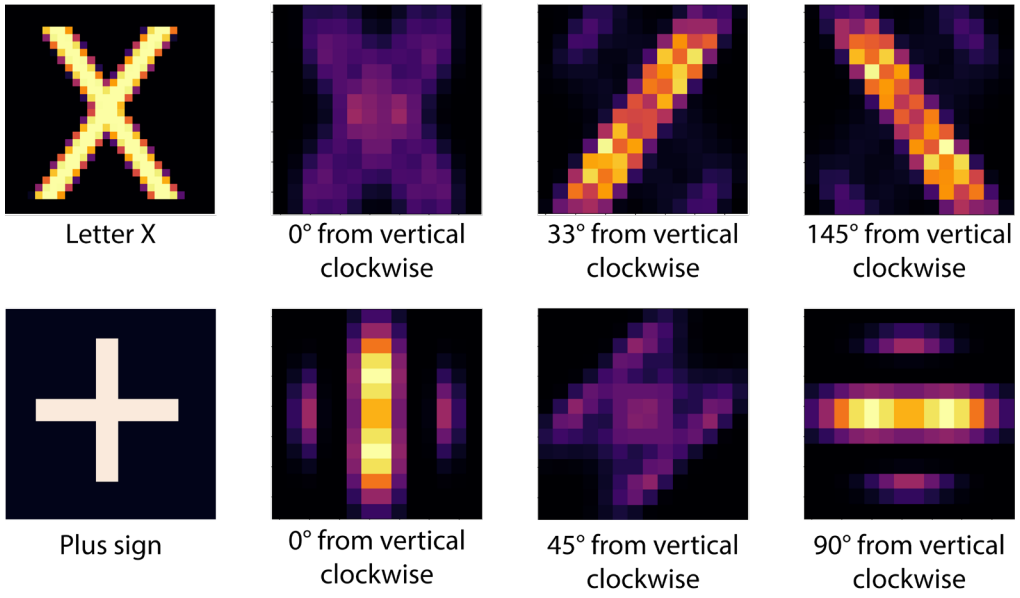
(h) Output tensor
Probability distribution (P)

(i)GlobalMaxPool

(j)Output

Translation invariant predictor stage

Scattering transform



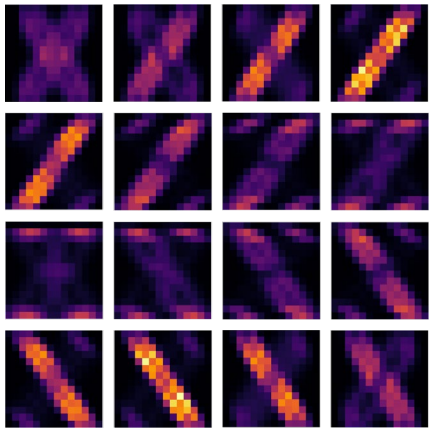
Scattering transform example wavelets

- A **wavelet scattering network** computes a translation invariant image representation which is stable to deformations and preserves high frequency information for classification.*
- Is an orthogonal transform based on wavelets that provides a roto-translational space.
- The angular sampling can be modified to achieve better angular sub-sampling.

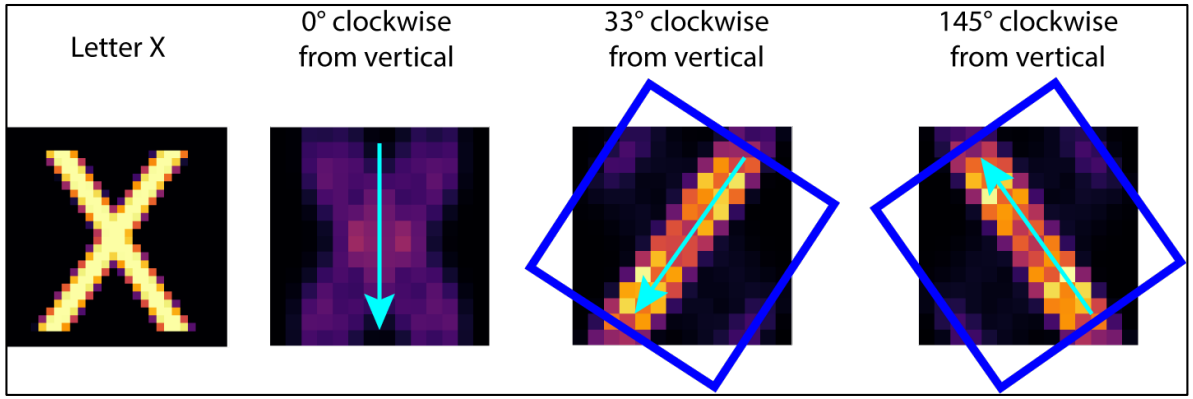
*** Stéphane Mallat, Joan Bruna.** *Invariant Scattering Convolution Networks.* CoRR, 2012.

Vectorization

An oriented scan order was implemented:



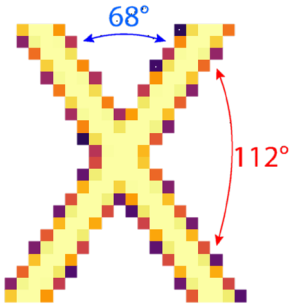
Scattering transform output
(M=2, J=1, L=16)



Custom scan order for angular features

- To achieve the invariance to the rotation the scan order of the images should have the same angle of the feature.
- This problem was solved by re-indexing the input data on custom bilinear scanning dense layer.

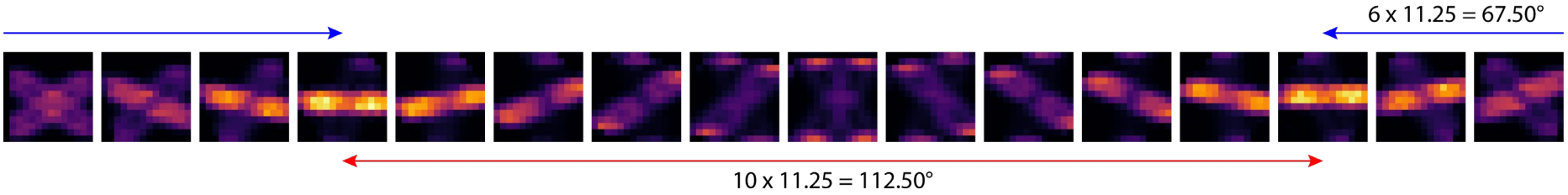
Oriented scan order wavelet behavior



Angles between letter X strokes

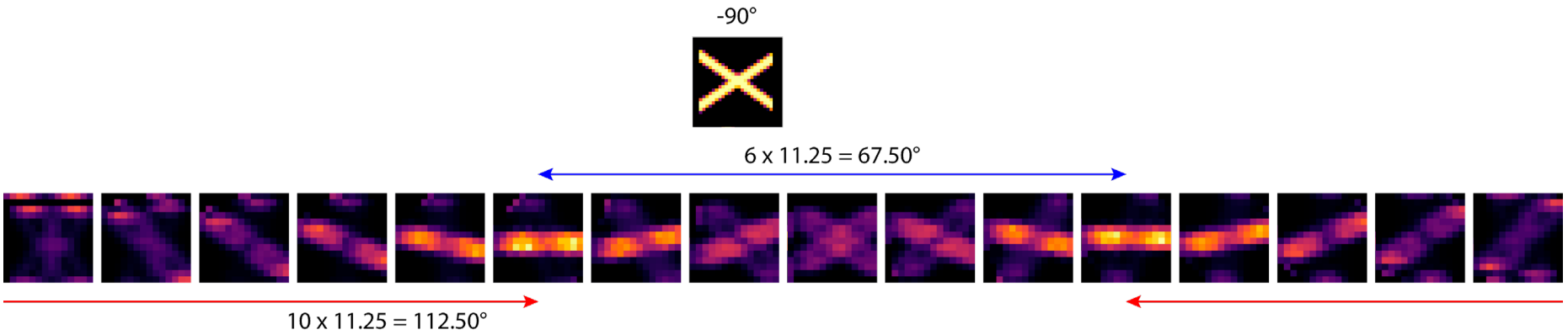
$$d\theta = \frac{180^\circ}{n} = \frac{180^\circ}{16} = 11.25^\circ$$

n = angular samples of scattering transform



- Letter X has two strokes separated by 112° and 68°
- The magnitude of this values is translated to linear separation between the angular filters.
- This magnitude remains constant for each rotation.

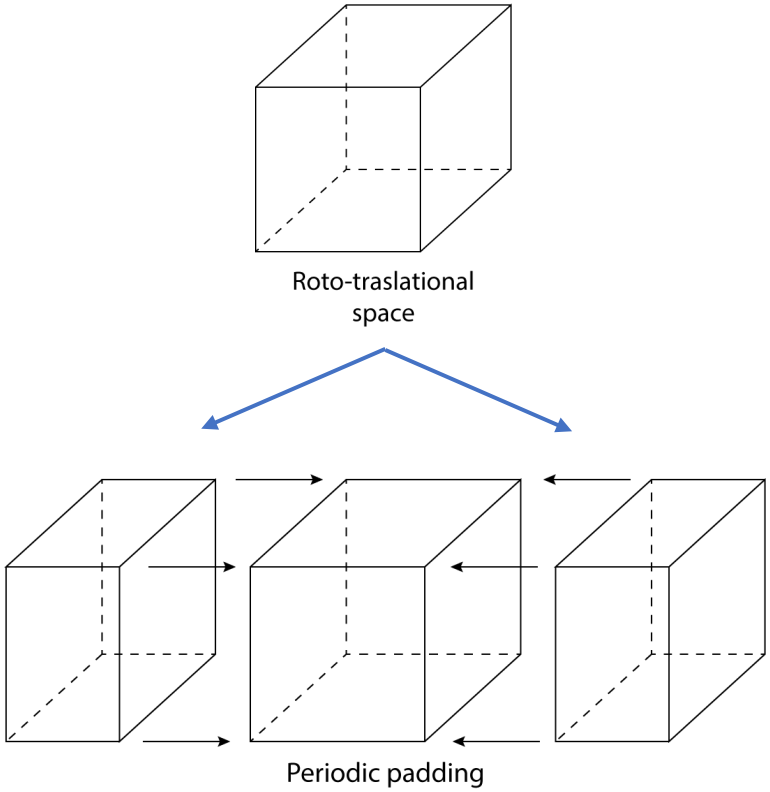
Oriented scan order wavelet behavior



Co-variance between angular input and shift over the output vector

- A covariance can be observed between the angular difference int the input image (letter X) and the translation over the channels.
- The angular information of the features in the input is preserved as the separation between channels and remains constant over different angular inputs.

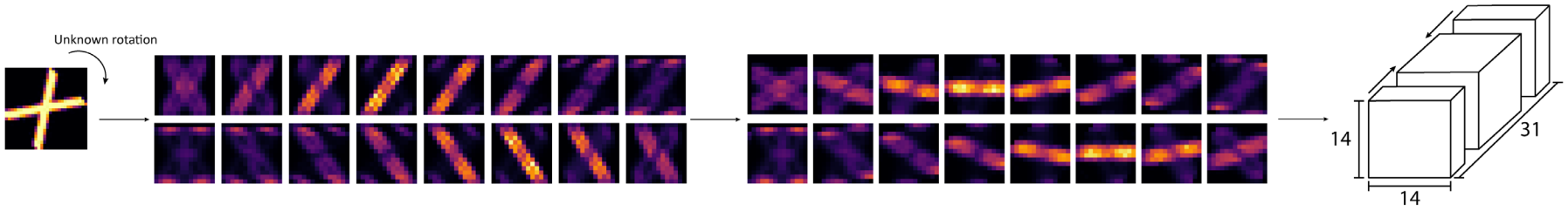
Periodic padding



A cyclic convolution can be implemented with a periodic padding of the roto-translational space by itself and a linear 3D convolution.

This padded space contains all the possible translations of the input.

Deep neural network proposal



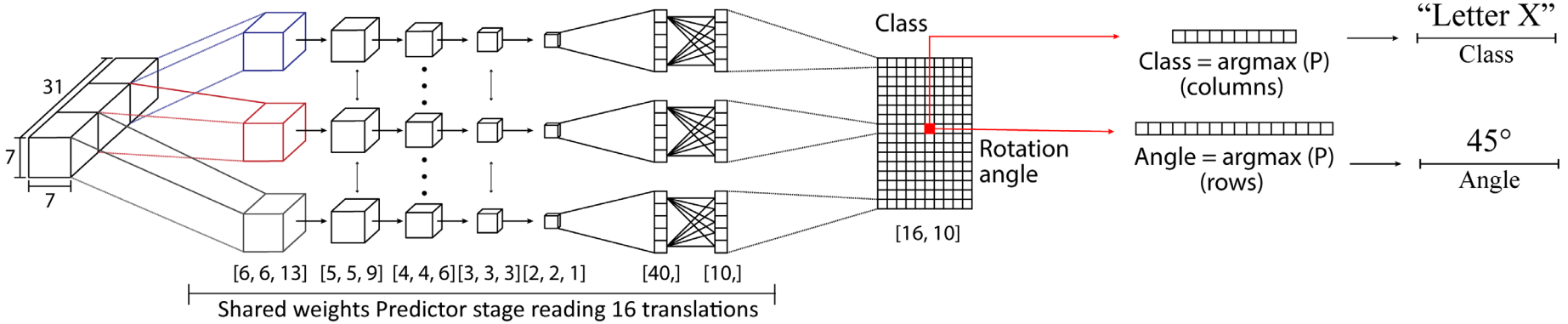
(a) Input pattern

(b) Oriented wavelet feature space
[14, 14, 16]

(c) Horizontally-oriented wavelets
[14, 14, 16]

(d) Periodic padding
[14, 14, 31]

Roto - translation covariance



(e) MaxPooling3D
[7, 7, 31]

(f) (5x) 3D Convolution
[2, 2, 1, 10]

(g) Dense Layer
[10,]

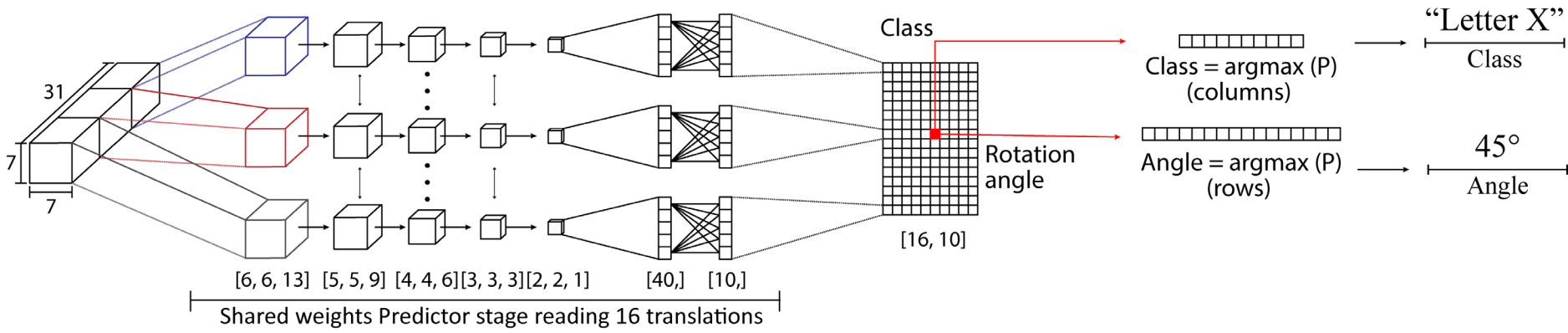
(h) Output tensor
Probability distribution (P)

(i)GlobalMaxPool

(j)Output

Translation invariant predictor stage

Oriented convolutional predictor



(e) MaxPooling3D [7, 7, 31]

(f) (5x) 3D Convolution [2, 2, 1, 10]

(g) Dense Layer [10,]

(h) Output tensor Probability distribution (P)

(i)GlobalMaxPool

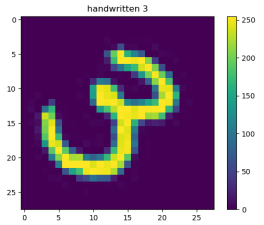
(j)Output

- A series of 3D Convolution predictors allow the network to see all rotated copies of the input pattern. This predictor reads each one of the possible orientations and generates a higher-class probability for the translation that contains the up-right orientation.

- A shared weights Dense layer scans over the output of the convolutions and outputs a prediction for each one of the translations in the form one out of many.

- The predictor have 7,022 trainable parameters.

Probability matrix



Input

Classes

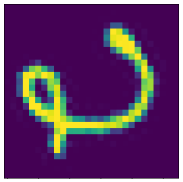
Angles

	0	1	2	3	4	5	6	7	8	9
0°	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.84	0.00
11°	0.00	0.00	0.00	0.72	0.00	0.00	0.00	0.00	0.96	0.00
22°	0.00	0.00	0.00	1.75	0.00	0.00	0.00	0.00	0.70	0.00
33°	0.00	0.00	0.00	2.40	0.00	0.00	0.00	0.00	0.42	0.00
44°	0.00	0.00	0.00	2.21	0.00	0.00	0.00	0.00	0.41	0.00
55°	0.00	0.00	0.00	1.22	0.00	0.00	0.00	0.00	0.39	0.00
66°	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.28	0.00
77°	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.18	0.00
88°	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00
-77°	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
-66°	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
-55°	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
-44°	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
-33°	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
-22°	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
-11°	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.19	0.00

Output matrix

- All the results from the convolutional predictor are condensed in a single matrix.
- In every column we should have only a maximum (predicted class maximum).
- The expected behavior on a successful training is to have good prediction values in the corresponding angle, upper and down rows.

Study cases results



Input rotated image
unlabeled angle

[class , angle]
[2 , +45°]

Output prediction

The proposed architecture comparison:

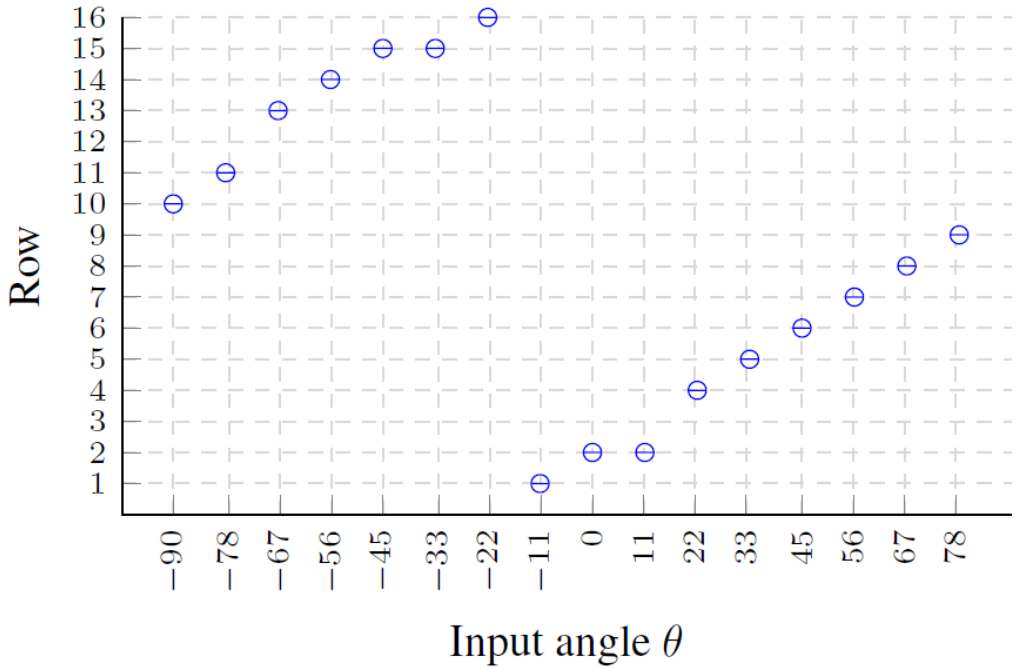
Upright samples training

Method	Error rate (in %)
SVM[1]	10.38±0.27
Harmonic Networks[2]	1.69
TI-Pooling[3]	1.2
Rot equiv. vector field networks[4]	1.09
ORN[5]	0.76
Rotationally Invariant Conv. Mod.[6]	3.51
Covariant CNN (Ours)	2.69

Randomly rotated samples training

Method	Error rate (in %)
ORN-8(ORPooling)[1]	16.67
ORN-8(ORAlign)[1]	16.24
RotInv Conv. (RP_RF_1)[2]	19.85
RotInv Conv. (RP_RF_1_32)[2]	12.20
Covariant CNN (Ours)	17.21

Results



The output exhibits a self-organizing behavior of mapping consecutive angular values as consecutive rows in the table.

This comes as a result of non-zero class probability on with maximum probability on and lower on the previous and next angular steps.

When the absolute angular reference is unknown (e.g. for plankton upright position does not exist) the network maps one of the rotation values to one point of the linear space and then the consecutive angles are linearly mapped.

Conclusions

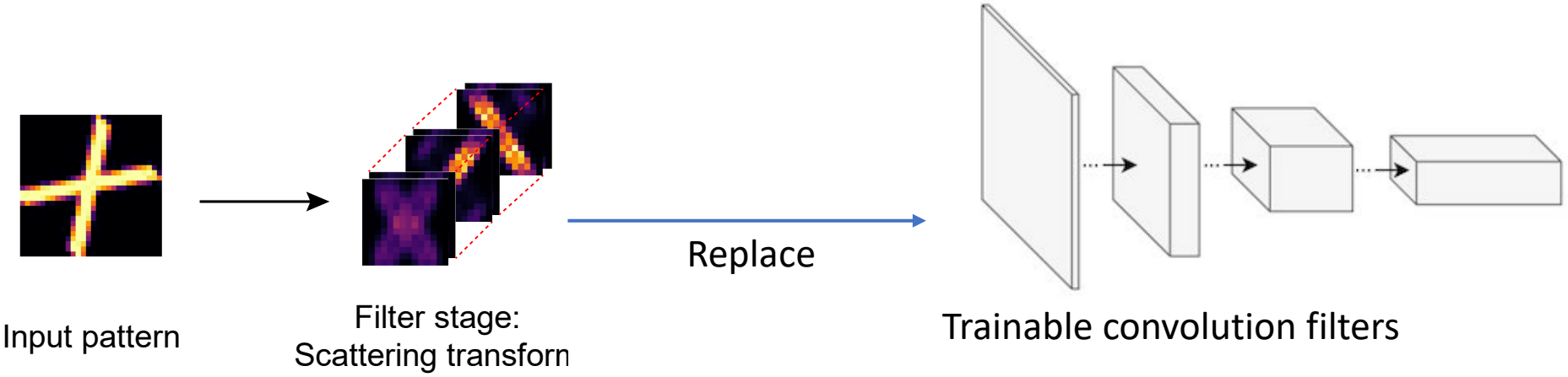
- A rotation-invariant deep learning architecture was presented.

- This architecture uses oriented features generated by a orthogonal transform based on wavelets scattering transform.

- The contributions of this architecture:
  1. A custom bilinear scanning dense layer by renumbering the order of the inputs to achieve invariance to the rotation.
  2. A rotation invariant predictor that outputs class and angle without angle labeling.
  3. The ability to train from up-right databases and predict over rotated samples without data augmentation techniques.

- Two main study cases of training on MNIST dataset as starting point were presented:
  1. Random oriented data: **2.69% error rate**
  2. Normally up-right data: **17.21% error rate**

Future works



- The scattering transform has the rotation invariant capabilities but is not able to fully describe the input, the next step is to generate a series of trainable rotation invariant convolution predictors.

- Test in other kind of datasets like food, plankton and rotated people faces.

Thank you for your attention!

Contact:
r.rodriguez@esiee.fr