Convolutional Neural Network with Structured Filters

Xiuyuan Cheng
Duke University

Mathematics Department, HKUST
June 2018

Outline

- Background
  - Convolutional neural network (CNN)
  - Network with pre-fixed weights
- Decomposed Convolutional Filters
  - DCF Net
  - Stability analysis
  - Experiments
- RotDCF: Group-Equivariant DCF Net
- Future Directions

Joint work with

- Qiang Qiu[1]

- Robert Calderbank[1,2]

- Guillermo Sapiro[1,2]

[1] Duke ECE [2] Duke Math

Outline

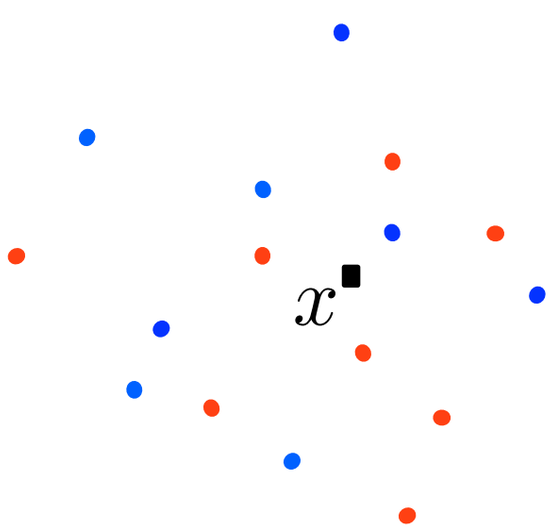
- **Background**
  - Convolutional neural network (CNN)
  - Network with pre-fixed weights
- Decomposed Convolutional Filters
  - DCF Net
  - Stability analysis
  - Experiments
- RotDCF: Group-Equivariant DCF Net
- Future Directions

Representation Learning

- To build a “good” representation

$$\text{Data: } x \in \mathbb{R}^d$$

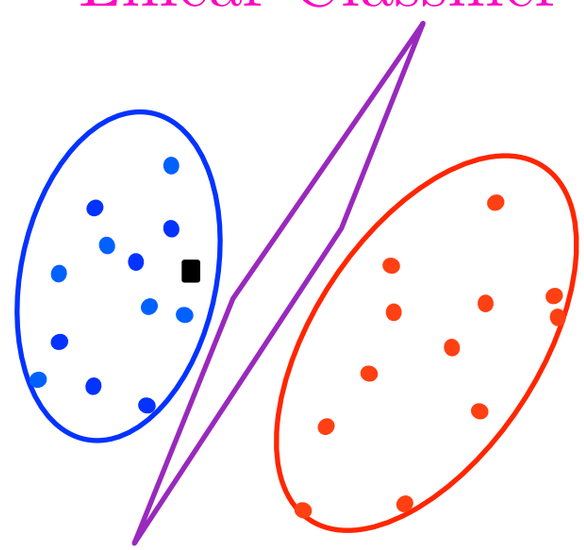
$$\|x - x'\| \text{: non-informative}$$



$$\Phi$$

$$\text{Representation } \Phi(x) \in \mathbb{R}^{d'}$$

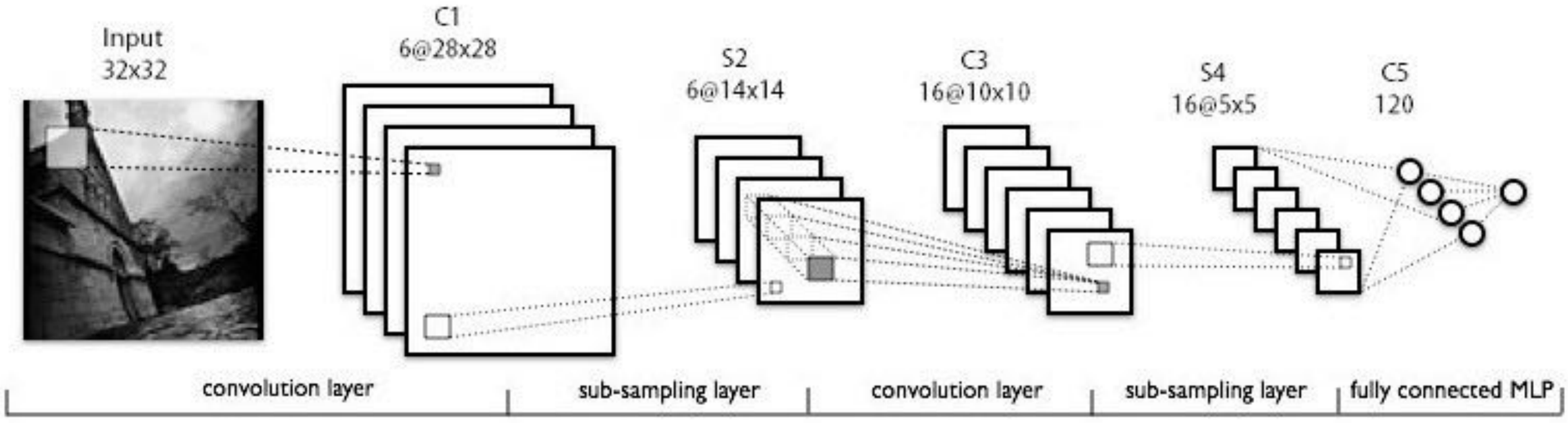
Linear Classifier



contractive & discriminative
(non-expansive)

Convolutional Neural Network

- Convolutional neural network (CNN)



Convolution + Nonlinear activation + “Pooling”

Convolutional Neural Network

- Image is lying on a **2D domain** ⟶ convolution

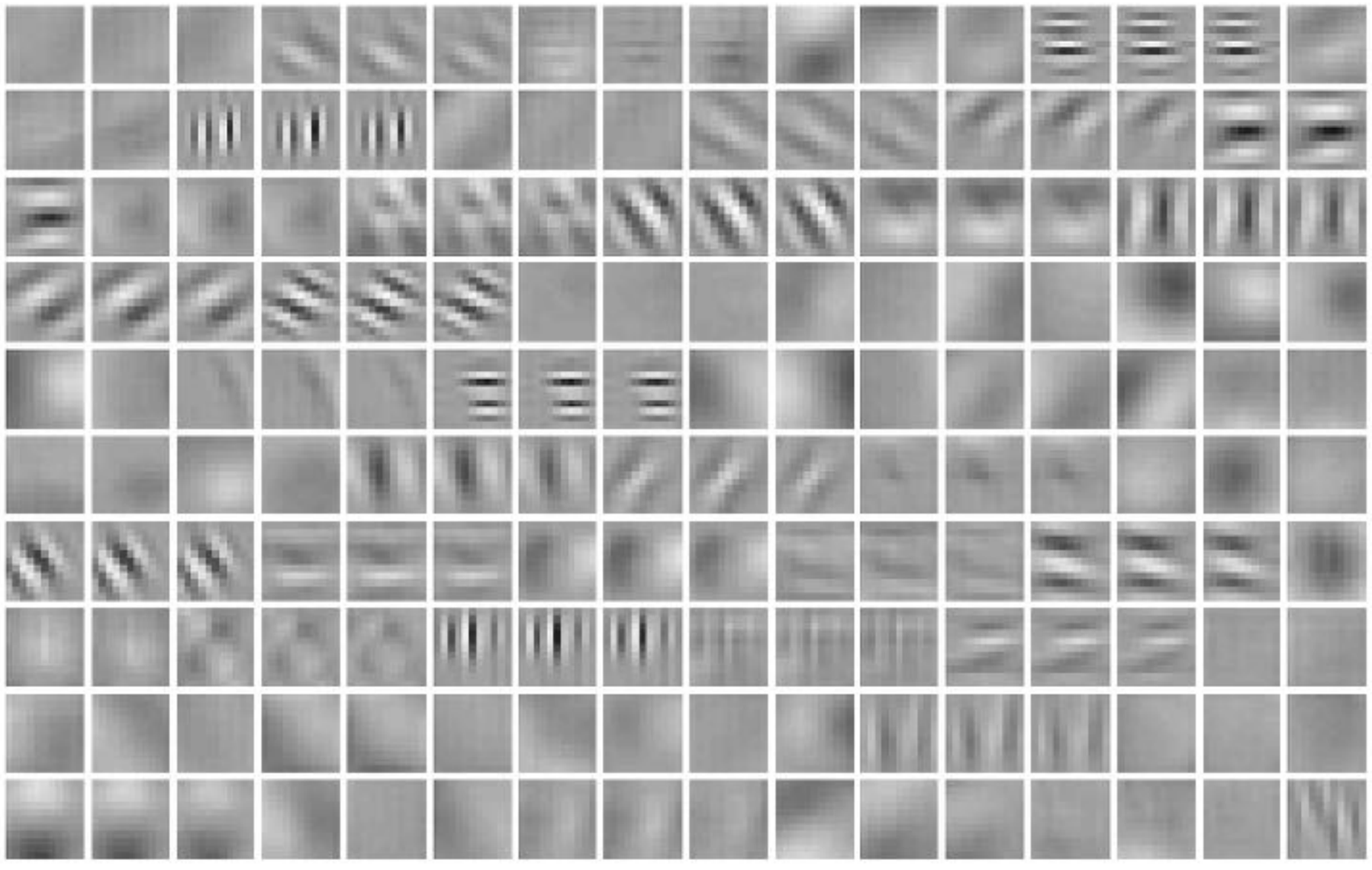
- Labels are insensitive to **nuance variations**

  ⟶ the same with the deep features

small translation, rotation, deformation,
change of scale/ brightness/color etc.

Convolutional Filters

- Typically trained from data in a supervised way

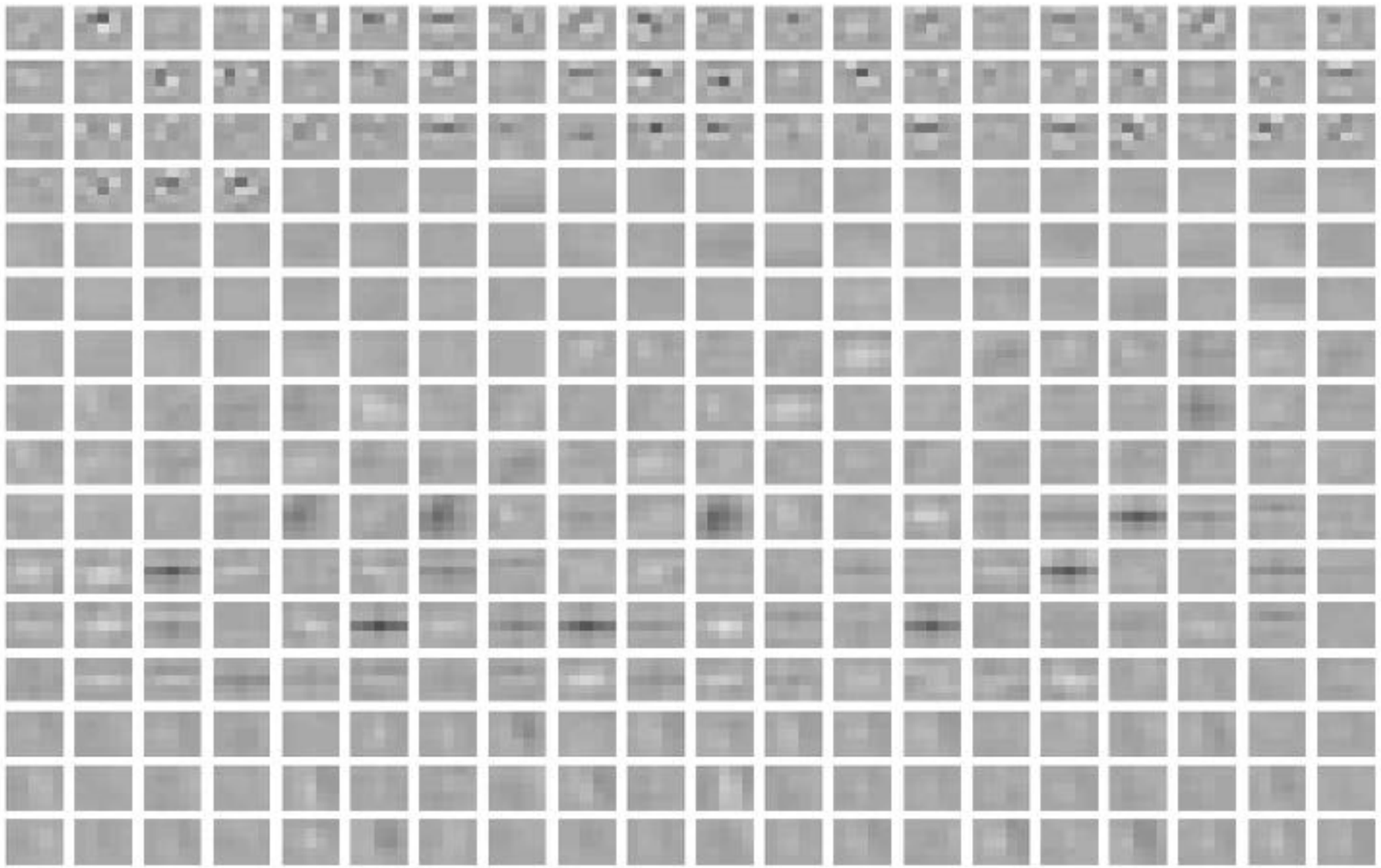


1st conv layer

Vgg-f net [Chatfield et al ‘14]

Convolutional Filters

- Typically trained from data in a supervised way

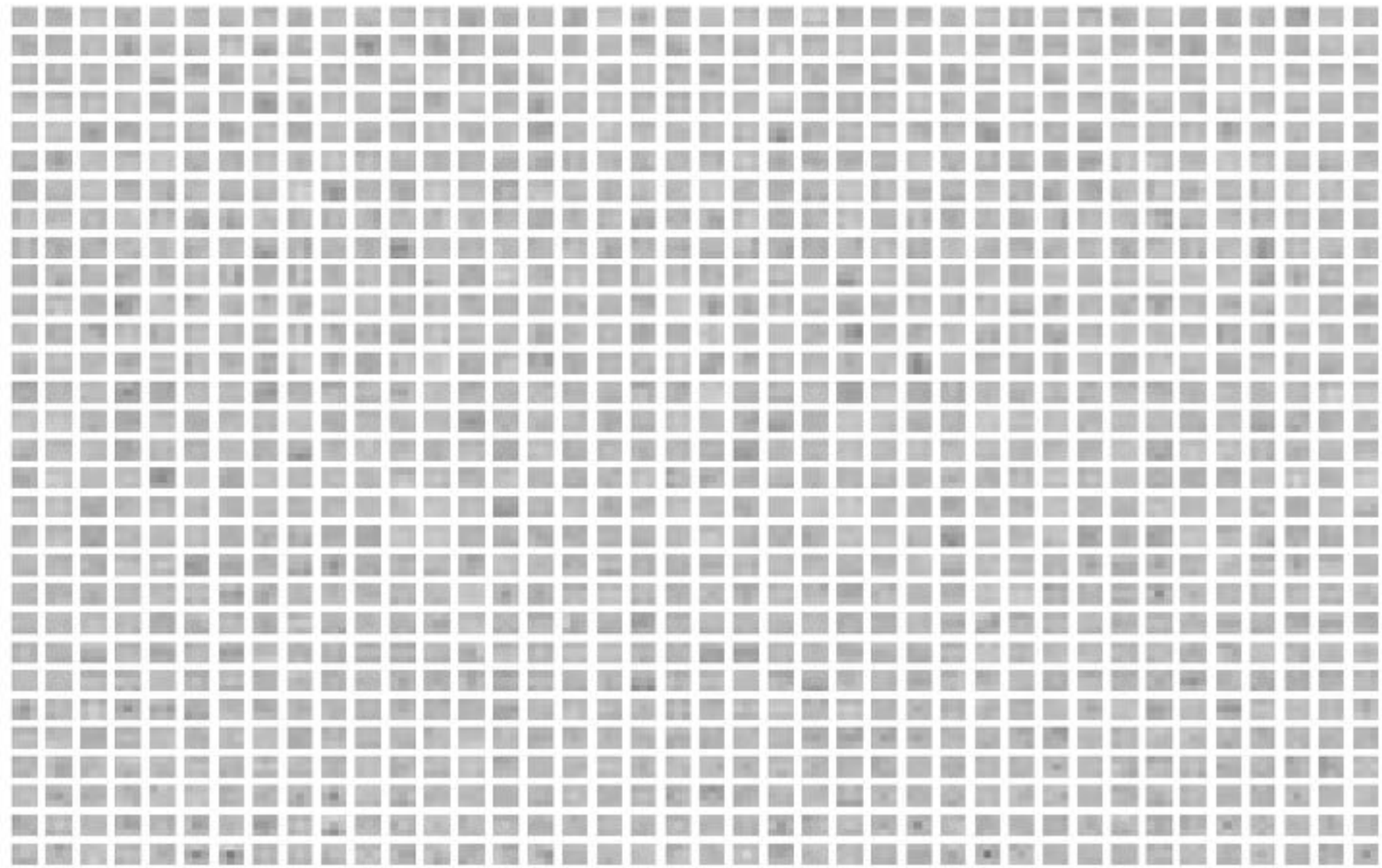


2nd conv layer

Vgg-f net [Chatfield et al ‘14]

Convolutional Filters

- Typically trained from data in a supervised way



Last conv layer

Vgg-f net [Chatfield et al ‘14]

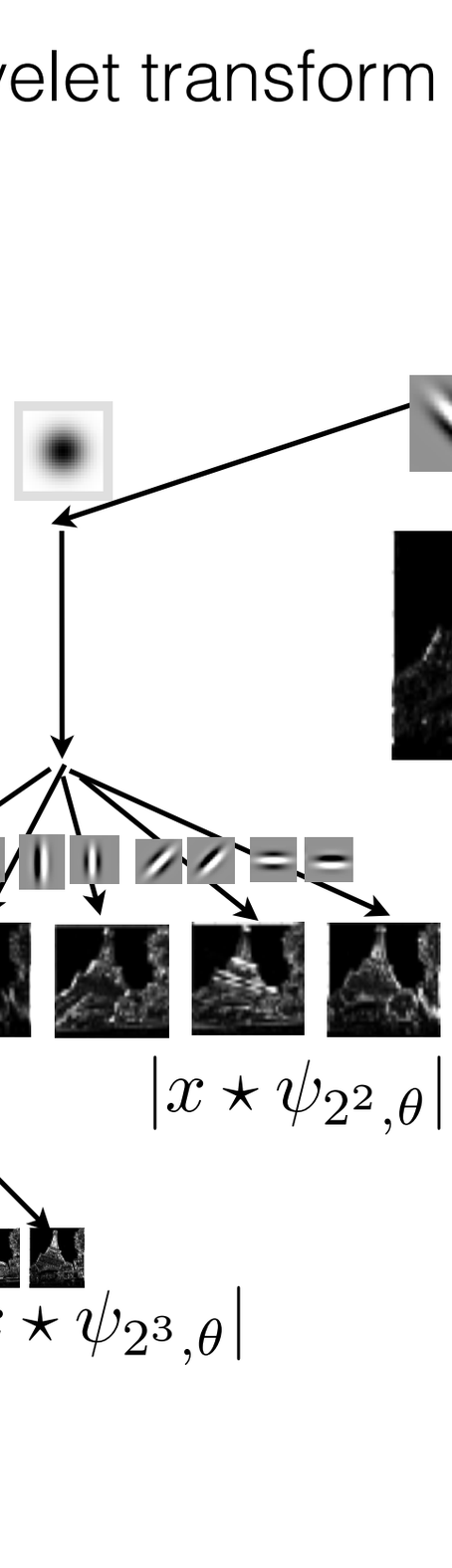
Convolutional Filters

- Supervised-ly trained filters
  1) Regularity over space
  2) Repetition and redundancy
- Alternatives?
  - Unsupervised filters:
    “PCA-net” [Chan et al.’14]
  - Pre-fixed (non-adaptive) filters like wavelets:
    “Scattering Transform” [Mallat ’12]

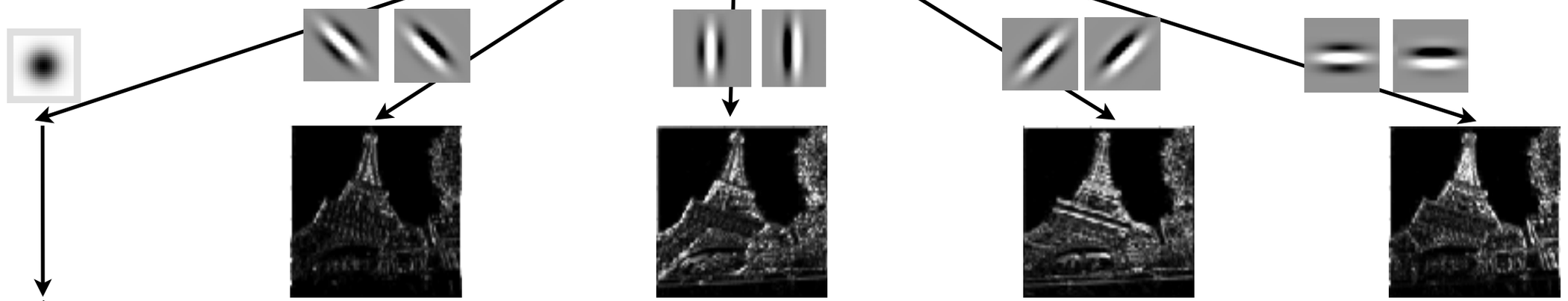
Scattering Networks

[Mallat '12]

Wavelet transform



x



$|x \star \psi_{2^1,\theta}|$

Scattering transform

$$S_J x = \begin{pmatrix} x * \phi_{2^J} \\ |x * \psi_{j_1,\theta_1}| * \phi_{2^J} \\ ||x * \psi_{j_1,\theta_1}| * \psi_{j_2,\theta_2}| * \phi_{2^J} \\ \cdots \end{pmatrix}_{j_1 < j_2 < \cdots < j_m \leq J}$$

2^0

2^1

2^2

2^3

2^J

Scale

Scattering Networks

[Mallat '12]

Stability of scattering representations

- Non-expansive mapping

$$\|S_J x - S_J y\| \leq \|x - y\|$$

- Deformation insensitivity

$$D_\tau x(u) = x(u - \tau(u)), \quad \|S_J D_\tau x - S_J x\| \leq C(\tau, J)\|x\|$$

No fitting,
Thus no overfitting!

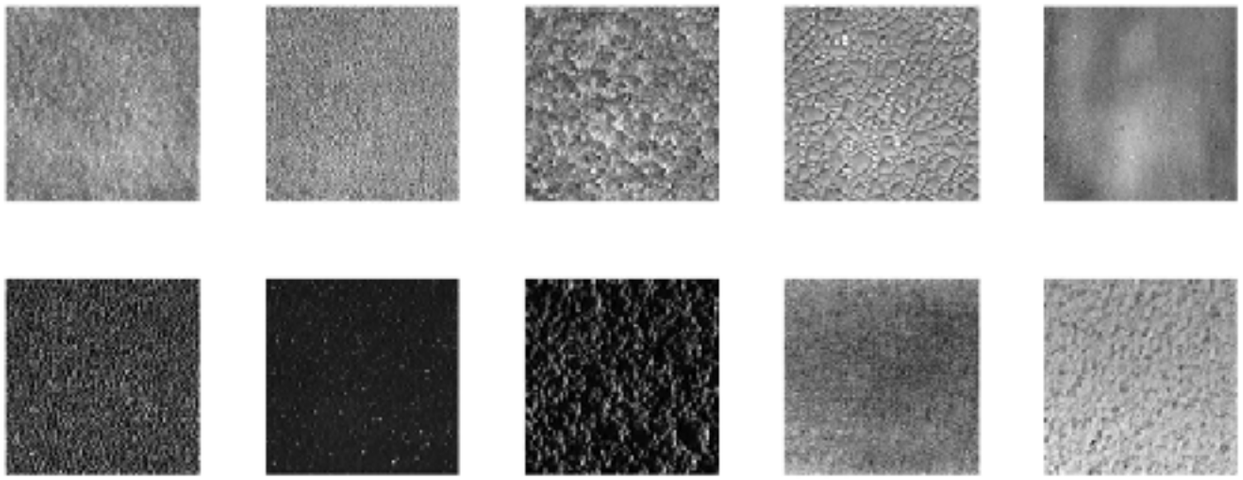
Scattering Networks

Applications and extensions:

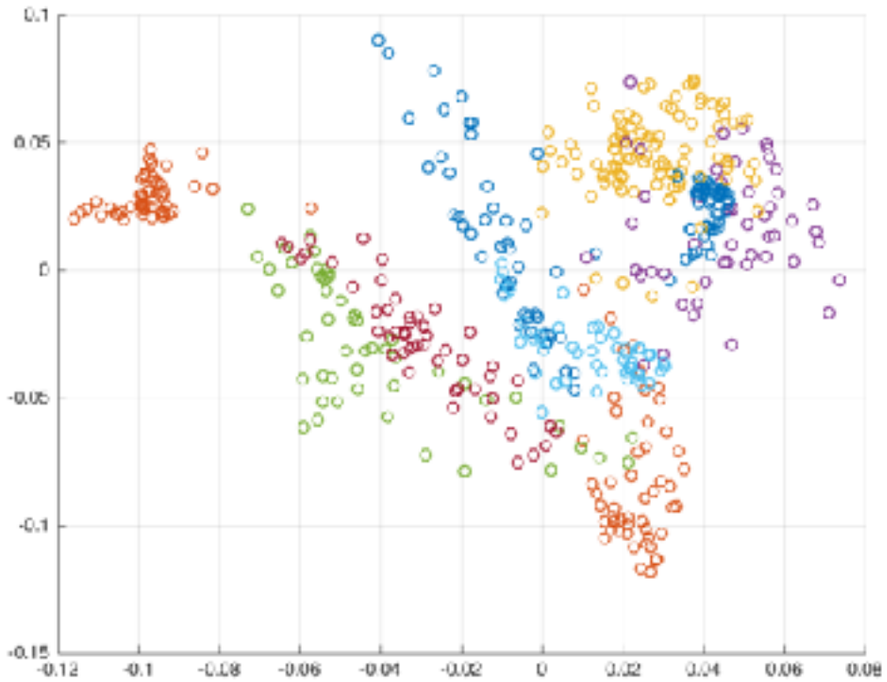
- Applications in image, texture, audio data [Andén & Mallat ’11] [Bruna & Mallat ’13] [Sifre & Mallat ’13]
- Invertibility/completeness of representation [Waldspurger et al. ’12]
- Extension to signals on graphs [Chen et al. ’14] [Cheng et al. ’16]
- With general family of filters [Bolcskei et al. ’15] [Czaja et al. ’15]
- (More)

Example: Haar Scattering

- Classification by scattering features using Haar wavelets



CuRET 10 Classes



Projected to 2 leading principal components

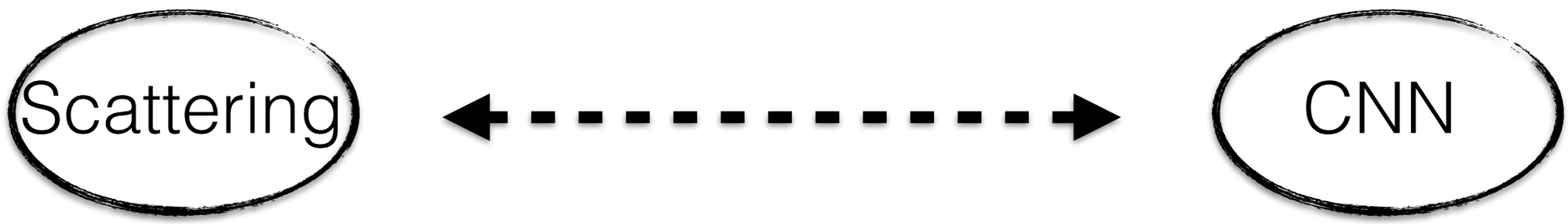
- Classification performance

	Haar Scat	**Gabor Scat**
CuRET (61 classes)	0.50%	0.20%
MNIST	0.59%	0.43%

Error rate

Background Summary

What is in between?



- No training until the classifier
- No parameters in the convolutional layers
- Most “control” of regularity and robustness
- Strong performance and explainable features

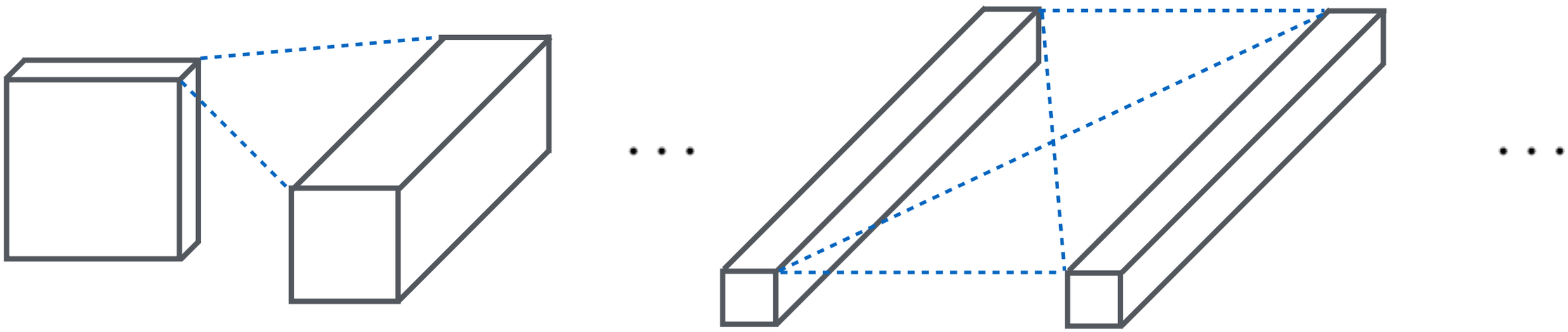
- Fully trained by large volume of data
- Lots of parameters (largest model capacity)
- Least “control” of regularity and robustness
- Best performance but not explainable

Outline

- Background
  - Convolutional neural network (CNN)
  - Network with pre-fixed weights
- Decomposed Convolutional Filters
  - **DCF Net**
  - Stability analysis
  - Experiments
- RotDCF: Group-Equivariant DCF Net
- Future Directions

Decomposition of Convolutional Filters

$$x^{(0)} \quad \mapsto \quad x^{(1)} \quad \mapsto \quad \cdots \quad \mapsto \quad x^{(l-1)} \quad \mapsto \quad x^{(l)} \quad \mapsto \quad \cdots$$



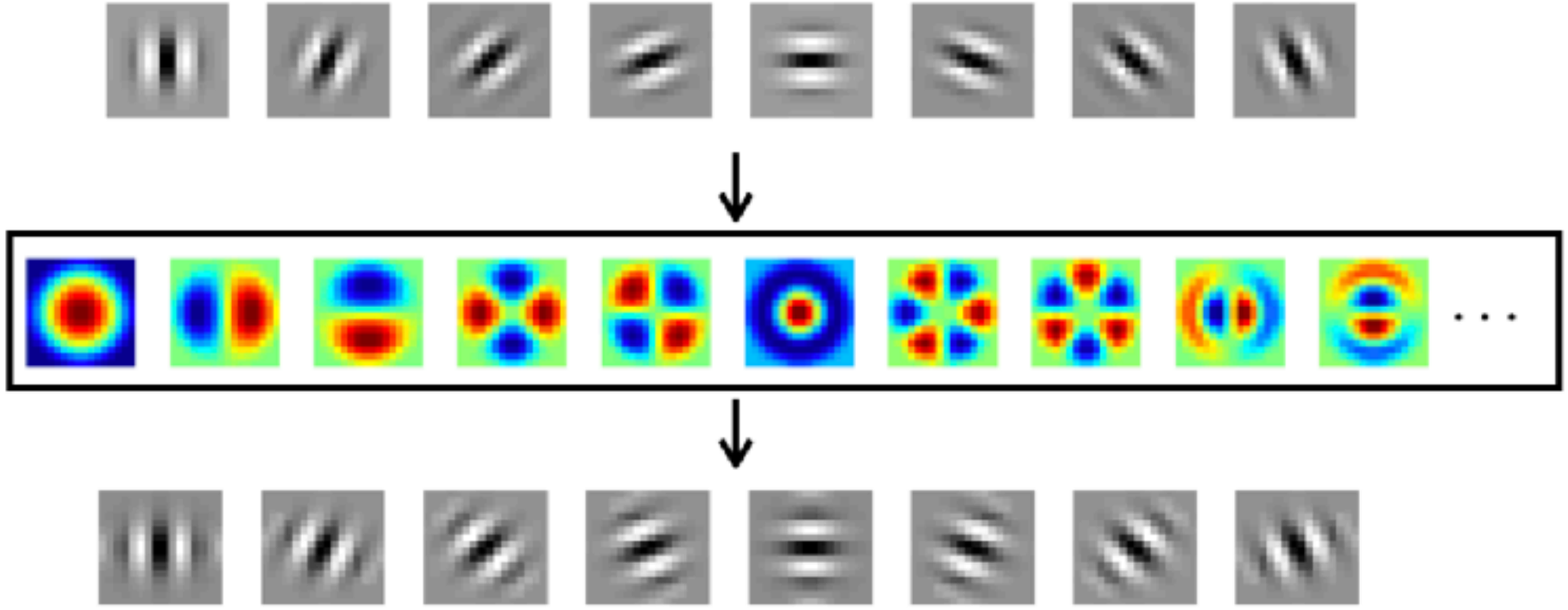
The mapping in a convolutional layer

$$x^{(l)}(u,\lambda) = \sigma\left(\sum_{\lambda'} \int W^{(l)}_{\lambda',\lambda}(v')x^{(l-1)}(u+v',\lambda')dv' + b^{(l)}(\lambda)\right)$$

Decomposition of Convolutional Filters

Introducing bases ψ_k

$$W_{\lambda',\lambda}(u) = \sum_{k=1}^{K} (a_{\lambda',\lambda})_k \psi_k(u).$$



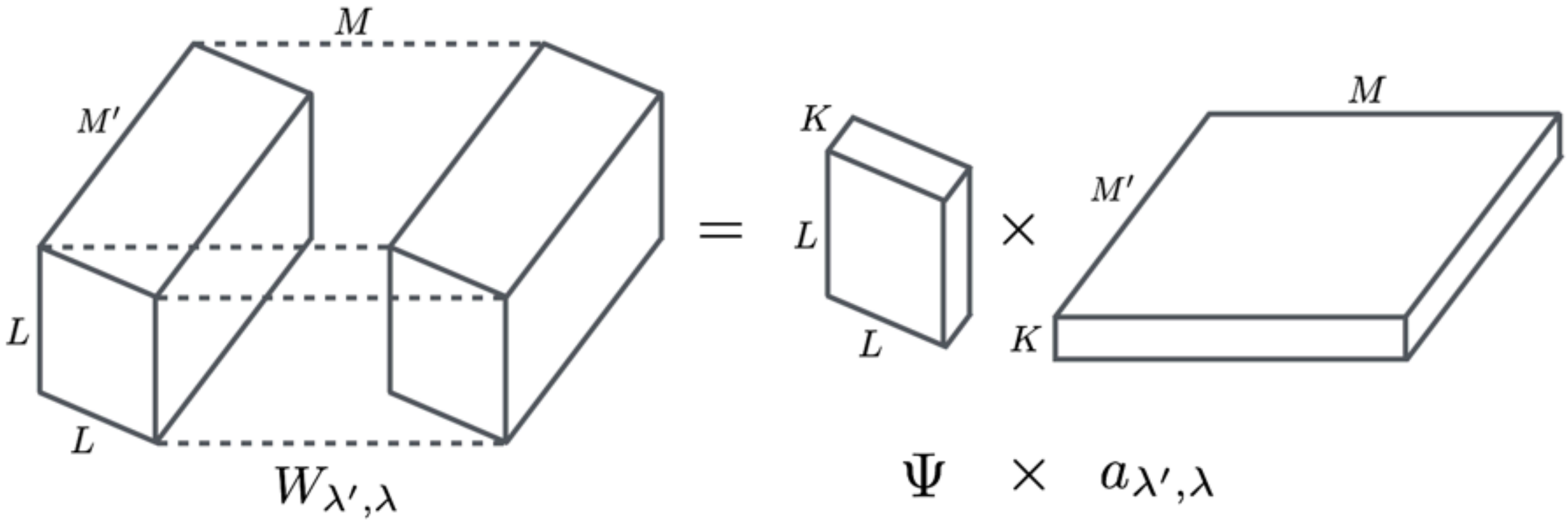
Decomposition of Convolutional Filters

A very natural idea, related works:

- Steerable filters (90's-): [Freeman et al '91], "steerable CNN" [Cohen & Welling '16]
- Sparse coding (00's-): "dictionary of dictionary" [Rubinstein et al. '10], "deep convolutional sparse coding" [Papyan et al. '16]
- Compression/pruning of deep networks (10's): "low-rank compression" [Denton et al '14], hashing and pruning [Chen et al. '15] [Han et al. '15 '16],"squeeze-net" [Iandola et al '17]
- Deep network with sparse connections (10's): [Changpinyo et al. '17] [Bolcskei et al' 17]
- (More)

Decomposition of Convolutional Filters

- Filters viewed in tensors



$W_{\lambda',\lambda}$ $\quad = \quad \Psi \quad \times \quad a_{\lambda',\lambda}$

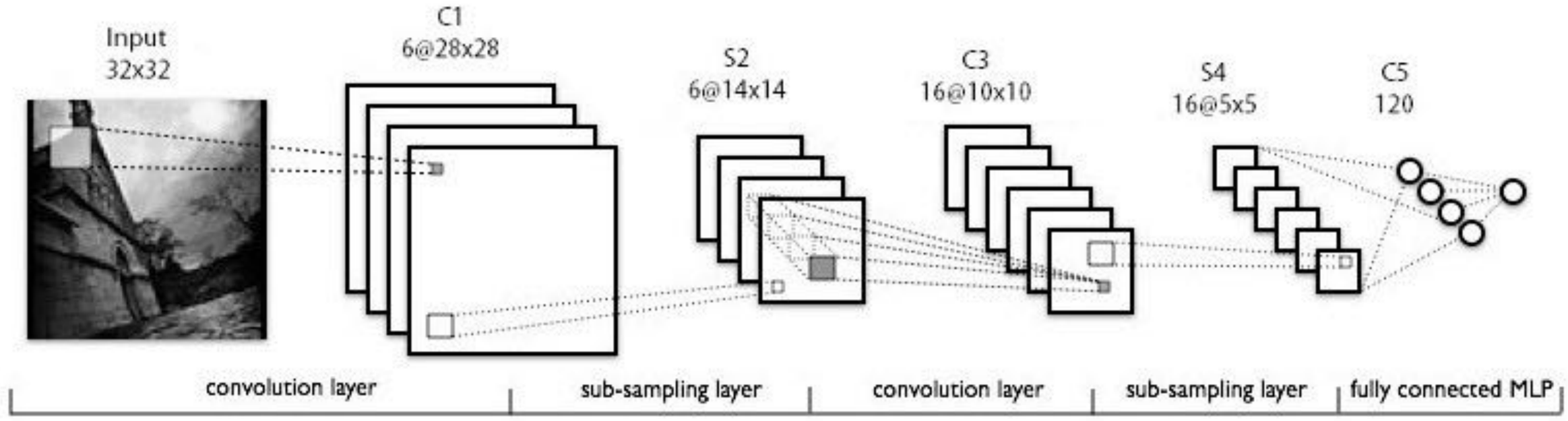
$[L, L, M', M]$ $\quad [L, L, 1, K]$ $\quad [1, 1, KM', M]$

- Psi prefixed, a trained from data

Multi-scale Convolutional Filters

Question: How to stack all the layers?

- Down-sampling (“pooling”) in CNN

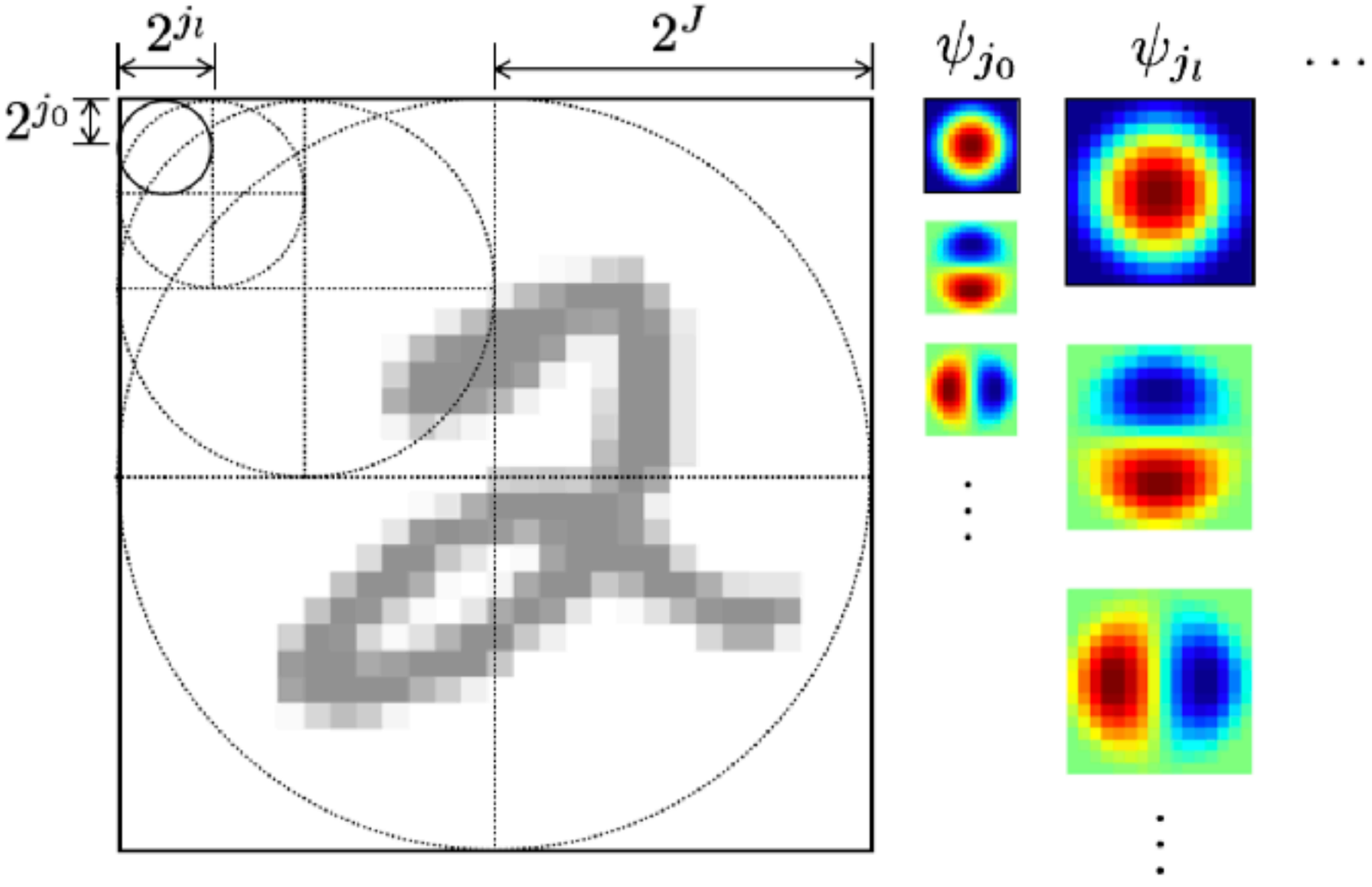


- Modeled by continuous convolution
  filters are rescaled by a factor of 2^j, each layer corresponds to some j

Multi-scale Convolutional Filters

- Rescale the bases accordingly

$$\psi_{j,k}(u) = 2^{-2j}\psi_k(2^{-j}u), \quad u \in D(j)$$



Reduction in the Number of Parameters

- Number of parameters
  - Regular conv layer: $L \times L \times M' \times M$
  - DCF layer: $K \times M' \times M$
- Forward-pass computation
  - Regular conv layer: $M'W^2 \cdot M(1 + 2L^2)$
  - DCF layer: $M'W^2 \cdot 2K(L^2 + M)$

A factor of $\frac{K}{L^2}$ *!*

Outline

- Background
  - Convolutional neural network (CNN)
  - Network with pre-fixed weights
- Decomposed Convolutional Filters
  - DCF Net
  - **Stability analysis**
  - Experiments
- RotDCF: Group-Equivariant DCF Net
- Future Directions

Representation Stability

Why stability?

An important theoretical problem:

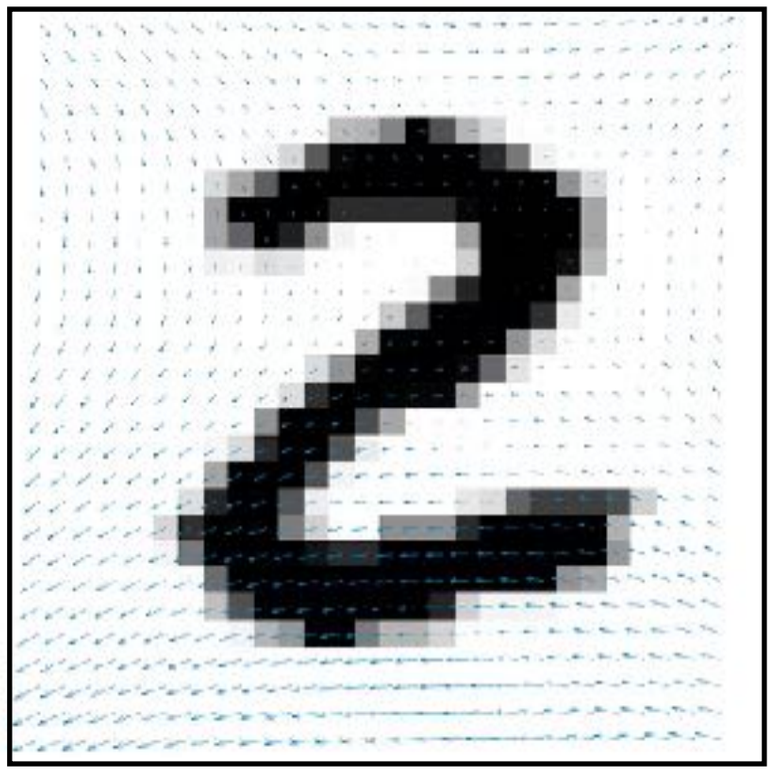
- Generalization of deep networks
- Adversarial examples

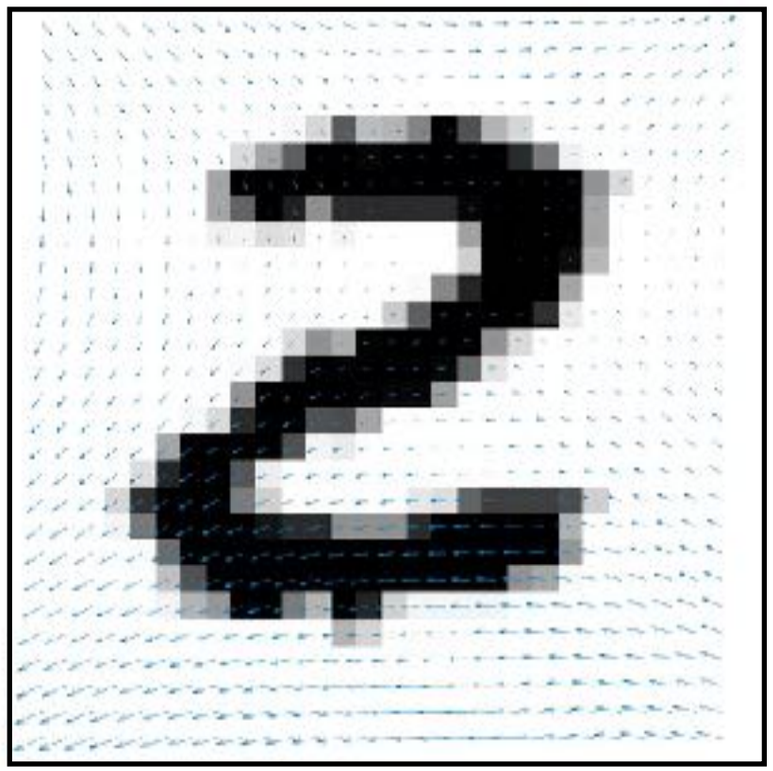
Related networks:

- Scattering and extension [Mallat ’12] [Wiatowski & Bolcskei ’15]
- Deep network with random weights [Giryes et al ’14]
- Convolutional kernel networks [Mairal et al ’14] [Bietti et al ’17]

Representation Stability

- Problem setting: smooth deformation





$$D_\tau x(u) = x(u - \tau(u)), \quad \tau : \mathbb{R}^2 \to \mathbb{R}^2$$

Representation Stability

- Goal of analysis

This needs to be reasonable!

$$\|x^{(L)}[x^{(0)}] - x^{(L)}[D_\tau x^{(0)}]\| \leq C(\tau, \text{network})\|x^{(0)}\|$$

- Assumptions

(A0) On the deformation:

$$|\nabla \tau|_\infty = \sup_u \|\nabla \tau(u)\| < \tfrac{1}{5}$$

(A1) On the nonlinear transform:

$$\sigma : \mathbb{R} \to \mathbb{R} \text{ is non-expansive}$$

(A2) On convolutional filters in a CNN:

$$\|W^{(l)}\|_1 \text{ property bounded, for all } l$$

Representation Stability

- Define technical constants for each layer

$$B_l := \max\{\sup_{\lambda} \sum_{\lambda'=1}^{M_{l-1}} \|W^{(l)}_{\lambda',\lambda}\|_1, \sup_{\lambda'} \frac{M_{l-1}}{M_l} \sum_{\lambda=1}^{M_l} \|W^{(l)}_{\lambda',\lambda}\|_1\}, \tag{1}$$

$$C_l := \max\{\sup_{\lambda} \sum_{\lambda'=1}^{M_{l-1}} \||v|\nabla W^{(l)}_{\lambda',\lambda}(v)\|_1, \sup_{\lambda'} \frac{M_{l-1}}{M_l} \sum_{\lambda=1}^{M_l} \||v|\nabla W^{(l)}_{\lambda',\lambda}(v)\|_1\},$$

- Assumption **(A2)** $B_l,\, C_l \leq 1$

1) To simplify the bounds
2) Normalization layers in practice

Representation Stability

- L^2 stability (non-expansiveness)

Proposition 1 *In a CNN, under (A1), if $B_l \leq 1$ for all l,*

(a) The mapping of the l-th convolutional layer (including σ), denoted as $x^{(l)}[x^{(l-1)}]$, is non-expansive, i.e., $\|x^{(l)}[x_1] - x^{(l)}[x_2]\| \leq \|x_1 - x_2\|$ for arbitrary x_1 and x_2.

(b) $\|x_c^{(l)}\| \leq \|x_c^{(l-1)}\|$ for all l, where $x_c^{(l)}(u, \lambda) = x^{(l)}(u, \lambda) - x_0^{(l)}(\lambda)$ is the centered version of $x^{(l)}$, $x_0^{(l)}$ being the output at the l-th layer from zero input. As a result, $\|x_c^{(l)}\| \leq \|x_c^{(0)}\| = \|x^{(0)}\|$.

Representation Stability

- Deformation stability: approximate **equivariant** relation

Lemma 2 *In a CNN, under (A0) (A1),* $c_1 = 4$,

$$\|D_\tau x^{(l)}[x^{(l-1)}] - x^{(l)}[D_\tau x^{(l-1)}]\| \leq c_1(B_l + C_l) \cdot |\nabla \tau|_\infty \|x_c^{(l-1)}\|. \qquad (1)$$

Proposition 3 *In a CNN, under (A0), (A1), (A2),* $c_1 = 4$,

$$\|D_\tau x^{(L)}[x^{(0)}] - x^{(L)}[D_\tau x^{(0)}]\| \leq 2c_1 L |\nabla \tau|_\infty \|x^{(0)}\|. \qquad (2)$$

- Bound $\|x^{(L)} - D_\tau x^{(L)}\|$, making use of the large-scale convolution with low-frequency filters.

- **Question**: How to satisfy **(A2)** $B_l,\ C_l \leq 1$ in a DCF Net?

Representation Stability

- Decomposition of convolutional filters under FB bases

$$W^{(l)}_{\lambda',\lambda}(u) = \sum_k (a^{(l)}_{\lambda',\lambda})_k \psi_{j_l,k}(u), \quad \psi_{j,k}(u) = 2^{-2j}\psi_k(2^{-j}u)$$

Proposition 4 *Let* $\|a\|_{FB} = (\sum_k \mu_k a_k^2)^{1/2}$. *Using FB bases,* $\||v||\nabla W^{(l)}_{\lambda',\lambda}(v)|\|_1$ *and* $\|W^{(l)}_{\lambda',\lambda}\|_1$ *are bounded by* $\pi\|a^{(l)}_{\lambda',\lambda}\|_{FB}$ *for all* λ', λ *and all* l.

- Assumption **(A2')** $A_l \leq 1$ for all layers

$$A_l := \pi \max\{\sup_\lambda \sum_{\lambda'=1}^{M_{l-1}} \|a^{(l)}_{\lambda',\lambda}\|_{FB}, \sup_{\lambda'} \frac{M_{l-1}}{M_l} \sum_{\lambda=1}^{M_l} \|a^{(l)}_{\lambda',\lambda}\|_{FB}\}$$

Representation Stability

- Deformation stability for DCF Net

Theorem 5 *In a DCFNet with FB bases, under (A0),(A1), (A2'),* $c_1 = 4$,

$$\|D_\tau x^{(L)}[x^{(0)}] - x^{(L)}[D_\tau x^{(0)}]\| \leq 2c_1 L|\nabla\tau|_\infty \|x^{(0)}\|.$$

Theorem 6 *In a DCFNet with FB bases, under (A0),(A1), (A2'),* $c_1 = 4$, $c_2 = 2$,

$$\|x^{(L)}[x^{(0)}] - x^{(L)}[D_\tau x^{(0)}]\| \leq (2c_1 L|\nabla\tau|_\infty + c_2 2^{-j_L}|\tau|_\infty)\|x^{(0)}\|. \quad (3)$$

1) The two terms match
2) Additive as L increases

Sketch of Proof

Key elements of the proof

- Relax by removing the sigmoid
- Change of variable into convolution with kernels
- Schur's test (Cauchy-Schwarz)
- Inserting bases expansion (rescaled to unit disk)

$$\|w\|_1, \||v||\nabla w(v)|\|_1 \leq \|\nabla w\|_1 \leq c\|\nabla w\|_2 \sim \|a\|_{\mathrm{FB}}$$

Outline

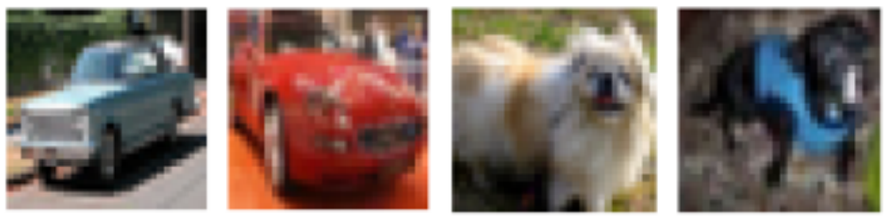
- Background
  - Convolutional neural network (CNN)
  - Network with pre-fixed weights
- Decomposed Convolutional Filters
  - DCF Net
  - Stability analysis
  - **Experiments**
- RotDCF: Group-Equivariant DCF Net
- Future Directions

Experiments

- Object recognition

MNIST

SVHN



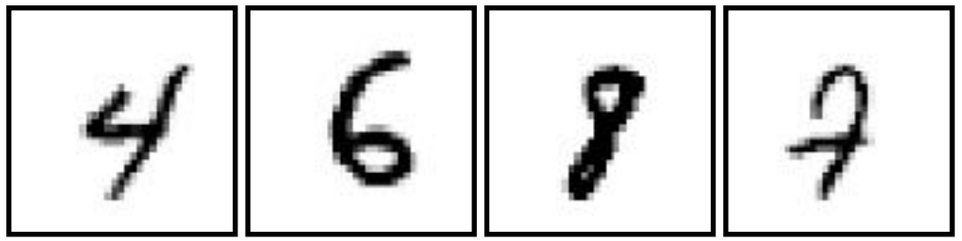
Cifar10

Conv-2
c5x5x1x16 ReLu mp3x3
c5x5x16x64 ReLu mp3x3
fc128 ReLu fc10

Conv-3
c5x5x3x64 ReLu mp3x3
c5x5x64x128 ReLu mp3x3
c5x5x128x256 ReLu mp3x3
fc512 ReLu fc10

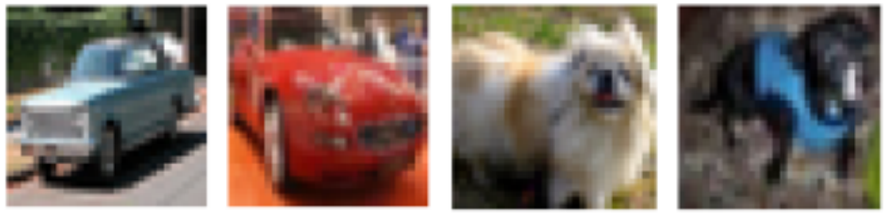
Experiments

- Object recognition



MNIST

SVHN

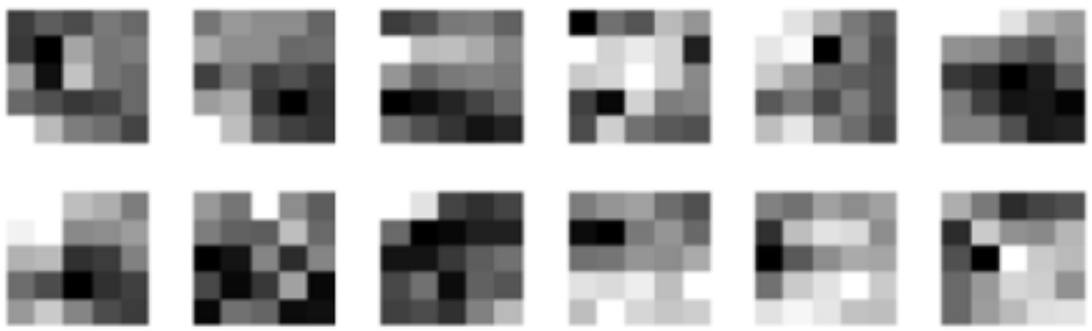


Cifar10

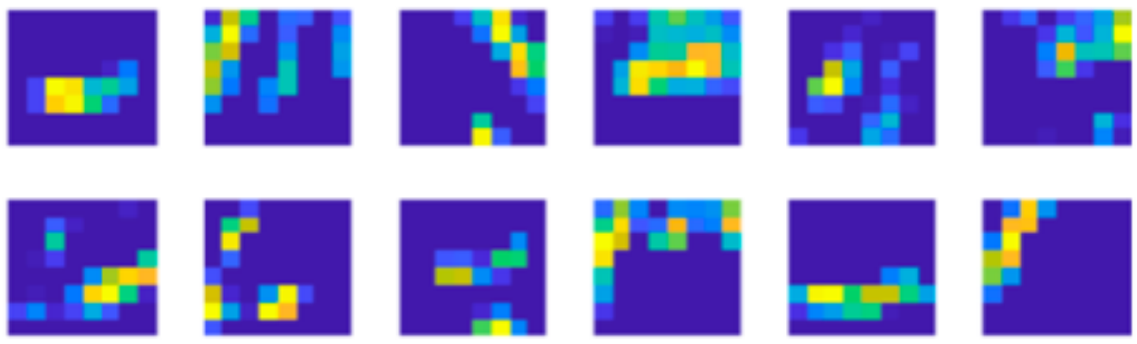
MNIST conv-2, 5x5						
	fb	rb	pca-s	pca-f	# param.	# MFlops
CNN	99.40				2.61×10^4	3.37
K=14	99.47	99.35	99.38	99.41	1.46×10^4	2.40
K=8	99.48	99.26	99.28	99.45	8.40×10^3	1.37
K=5	99.39	99.28	99.28	99.43	5.28×10^3	0.86
K=3	99.40	98.69	99.19	99.35	3.20×10^3	0.51
SVHN conv-3, 5x5						
	fb	rb	pca-s	pca-f	# param.	# MFlops
CNN	94.22				1.03×10^6	201.64
K=14	94.63	93.75	94.52	94.42	5.74×10^5	121.91
K=8	94.39	92.05	93.85	94.30	3.30×10^5	69.67
K=5	93.93	91.28	92.34	94.03	2.06×10^5	43.55
K=3	92.84	88.47	91.88	93.10	1.24×10^5	26.13
Cifar10 conv-3, 5x5						
	fb	rb	pca-s	pca-f	# param.	# MFlops
CNN	85.66					
K=14	85.88	84.76	85.27	85.34		
K=8	85.30	81.27	84.70	85.09	(same as above)	
K=5	84.35	77.96	83.12	83.94		
K=3	83.12	74.05	80.94	82.91		
Cifar10 vgg-16, 3x3						
	fb	rb	pca-s	pca-f	# param.	# MFlops
CNN	87.02				1.47×10^7	547.20
K=5	87.79	84.16	87.98	87.60	8.18×10^6	311.68
K=3	88.21	78.46	87.45	87.54	4.91×10^6	187.02

Experiments

Regular CNN

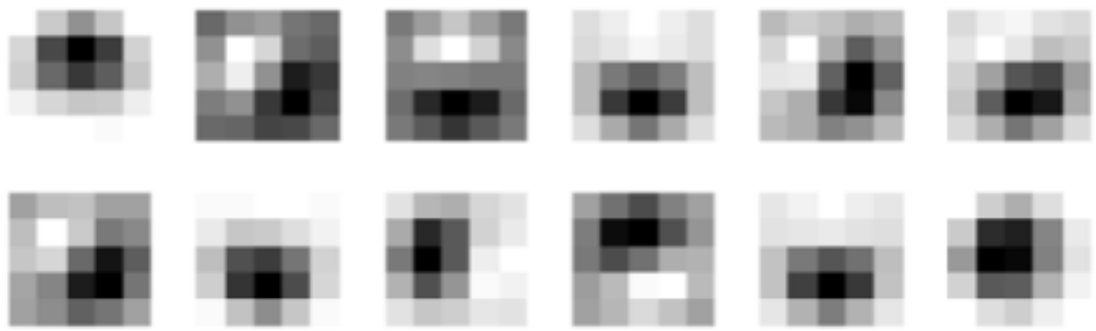


Trained conv filters

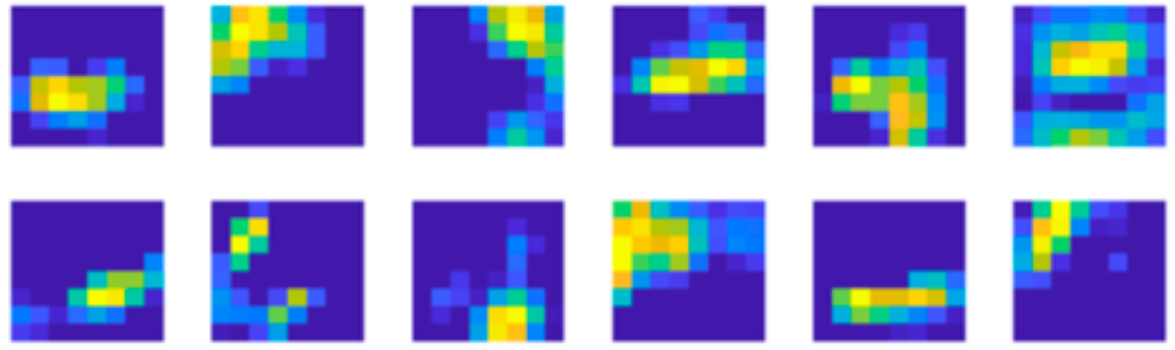


2nd layer output

DCF Net (K=3)



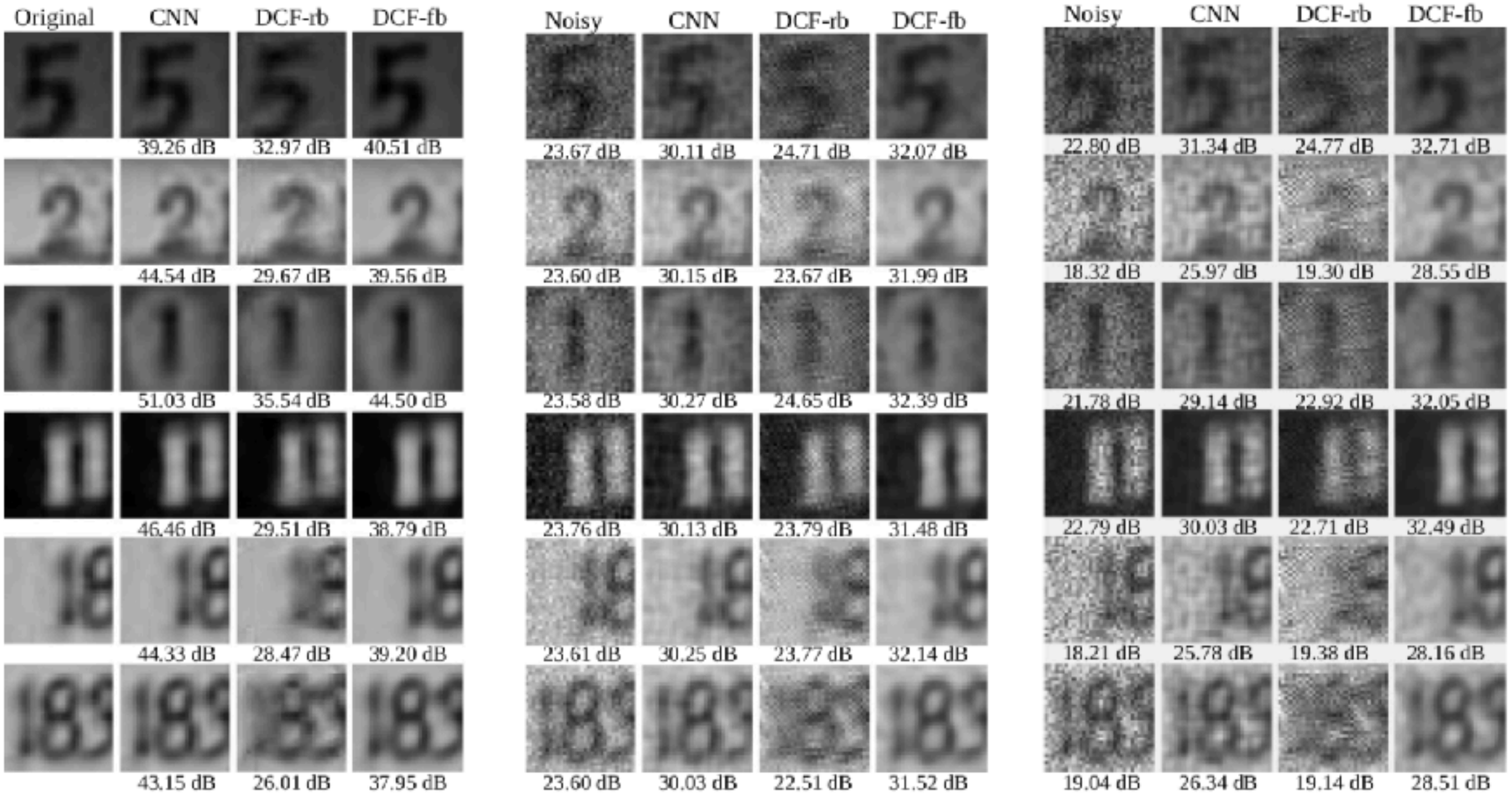
Trained conv filters



2nd layer output

Experiments

- Image reconstruction: network trained on clean images



(a) Original (b) Gaussian noise (c) Speckle noise

Mean PSNR over 20K testing set:

Gaussian noise: CNN 30.01, DCF-FB 31.24. Speckle noise: CNN 28.15, DCF-FB 29.84

Experiments

- Face verification: 2.6M face images from >2.6K people, LFW benchmark

Layer	CNN	DCFNet
1	conv $3 \times 3 \times 3 \times 64$	3 3 $\times$ 3 basis conv $1 \times 1 \times 9 \times 64$
2	ReLu	
3	conv $3 \times 3 \times 64 \times 64$	3 3 $\times$ 3 basis conv $1 \times 1 \times 192 \times 64$
4-5	ReLu, maxPool 2×2	
6	conv $3 \times 3 \times 64 \times 128$	3 3 $\times$ 3 basis conv $1 \times 1 \times 192 \times 128$
7	ReLu	
8	conv $3 \times 3 \times 128 \times 128$	3 3 $\times$ 3 basis conv $1 \times 1 \times 384 \times 128$
9-10	ReLu, maxPool 2×2	
(1-31 CNN layers are identical to *vgg-face* model in (Parkhi et al., 2015).)		
32	conv $5 \times 5 \times 512 \times 512$	8 5 $\times$ 5 basis conv $1 \times 1 \times 4096 \times 512$
33-34	ReLu, dropout	
35	conv $3 \times 3 \times 512 \times 512$	3 3 $\times$ 3 basis conv $1 \times 1 \times 1536 \times 512$
36-39	ReLu, dropout, FC, softmax	

	Accuracy	# param.	# GFlops
VGG-face	97.27 %	-	-
CNN	97.65 %	21.26×10^6	30.05
DCFNet	97.32 %	7.01×10^6	10.09

VGG-face accuracy [Parkhi et al ’15]

Outline

- Background
  - Convolutional neural network (CNN)
  - Network with pre-fixed weights
- Decomposed Convolutional Filters
  - DCF Net
  - Stability analysis
  - Experiments
- **RotDCF: Group-Equivariant DCF Net**
- Future Directions

RotDCF Network

- Rotation-equivariant CNN[1,2,3,4]

fully-connected layer	regular convolutional layer	CNN with group-indexed channels
$x^{(l-1)}(\lambda') \to x^{(l)}(\lambda)$	$x^{(l-1)}(u', \lambda') \to x^{(l)}(u, \lambda)$	$x^{(l-1)}(u', \alpha', \lambda') \to x^{(l)}(u, \alpha, \lambda)$
$\lambda' \to \lambda$: dense	$u' \to u$: spatial convolution $\lambda' \to \lambda$: dense	$u' \to u$, $\alpha' \to \alpha$: joint convolution $\lambda' \to \lambda$: dense

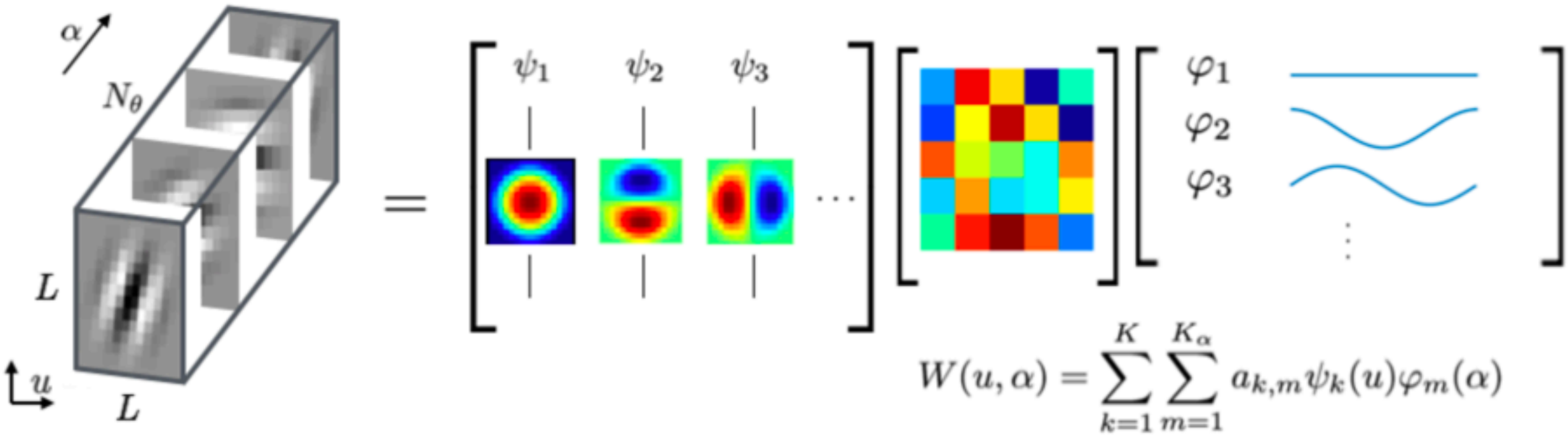
- Joint convolution over **space** and **group**

$$x^{(l)}(u, \alpha, \lambda) = \sigma\left(\sum_{\lambda'=1}^{M_{l-1}} \int_{S^1} \int_{\mathbb{R}^2} x^{(l-1)}(u + v', \alpha', \lambda') W^{(l)}_{\lambda', \lambda}(\Theta_\alpha v', \alpha' - \alpha) dv' d\alpha' + b^{(l)}(\lambda)\right)$$

[1] Cohen & Welling ’16 [2] Worrall et al ’17 [3] Zhou et al ’17 [4] Weiler et al ’17

RotDCF Network

- “Steerable filters” by steerable bases



- Rotation-equivariant CNN with **low model complexity** and **proved representation stability**

Experiments on Object Recognition

rotMNIST Conv-3, $N_{\text{tr}} = 10K$			
	Test Acc.	# param.	Ratio
CNN M=32	95.67	2.570×10^5	1.00
DCF M=32, K=5	95.58	5.158×10^4	0.20
DCF M=32, K=3	95.69	3.104×10^4	0.12
RotDCF $N_\theta = 8$			
M=16, K=14, K_α=8	97.86	2.871×10^5	1.12
M=16, K=5, K_α=8	97.81	1.026×10^5	0.40
M=16, K=3, K_α=8	97.77	6.160×10^4	0.24
M=16, K=5, K_α=5	97.96	6.419×10^4	0.25
M=16, K=3, K_α=5	97.95	3.856×10^4	0.15
M=8, K=5, K_α=5	97.81	1.610×10^4	0.06
M=8, K=3, K_α=5	97.59	9.680×10^3	0.04

rotMNIST Conv-3, $N_{\text{tr}} = 5K$			
	Test Acc.	# param.	Ratio
CNN M=32	94.04		
DCF M=32, K=3	94.08		
RotDCF N_θ=8			
M=16, K=3, K_α=5	96.79	(same as left)	
M=8, K=3, K_α=5	96.53		
CIFAR10 VGG-16, $N_{\text{tr}} = 10K$			
CNN $M = 64$	78.40	2.732×10^6	1.00
RotDCF, N_θ= 8			
M=32, K=3, K_α=7	79.44	1.593×10^6	0.58
M=32, K=3, K_α=5	79.53	1.138×10^6	0.42

- Higher **accuracy** due to group equivariance
- Low model **complexity** due to bases decomposition

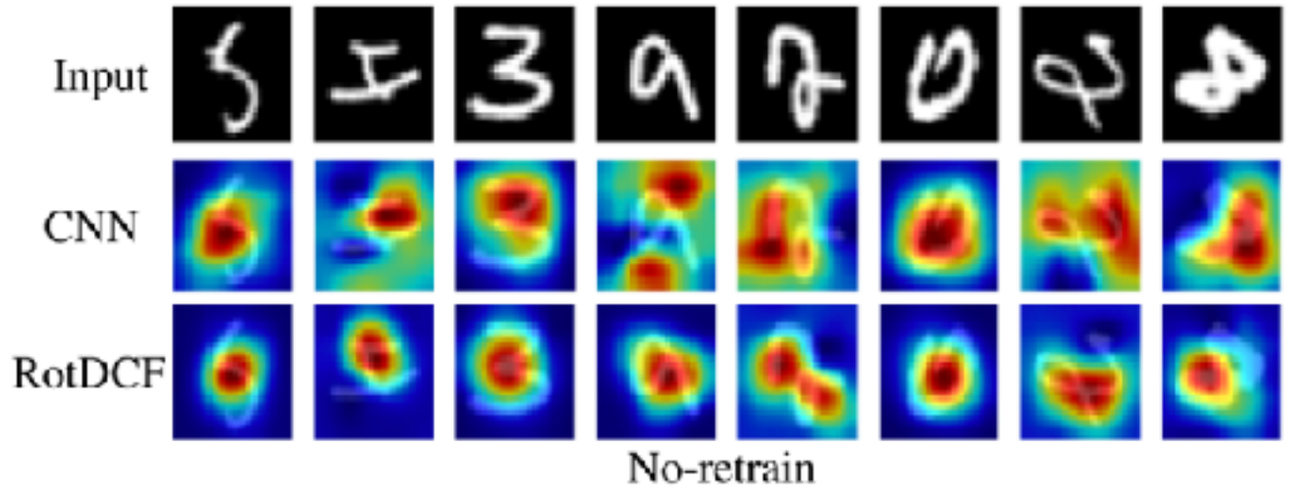
Transfer Learning to rotMNIST

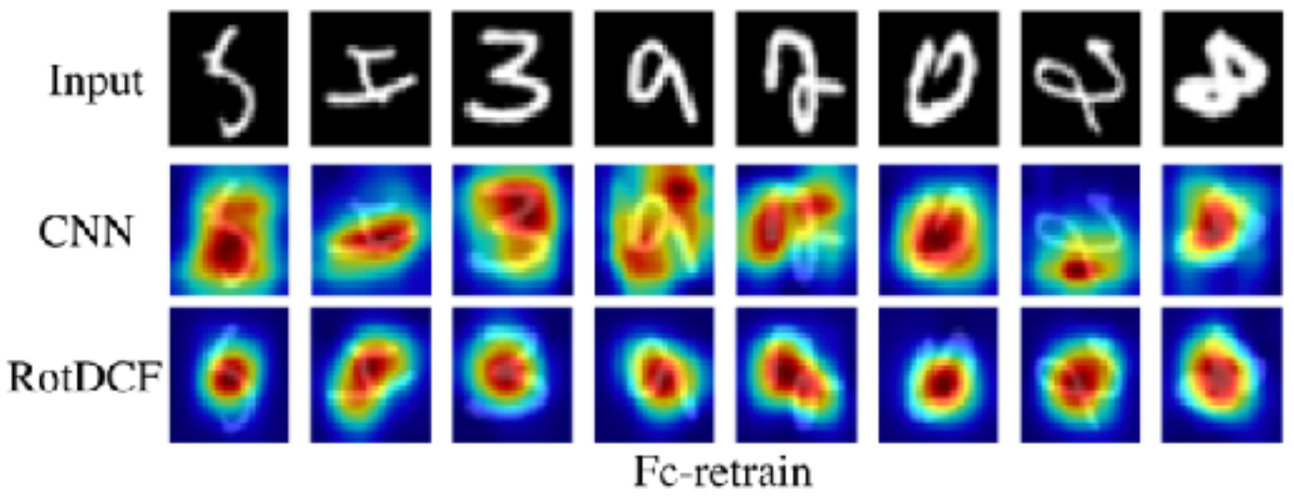
- Trained on 10K up-right digits and tested on 50K randomly rotated digits

MNIST to rotMNIST MaxRot=30 Degrees		
	no-retrain	fc-retrain
CNN	92.61	94.71
RotDCF	96.90	98.48

MNIST to rotMNIST MaxRot=60 Degrees		
	no-retrain	fc-retrain
CNN	69.61	85.90
RotDCF	82.36	97.68

- Class activation maps (CAM)[1] on testing images, maxRot=60 degrees.

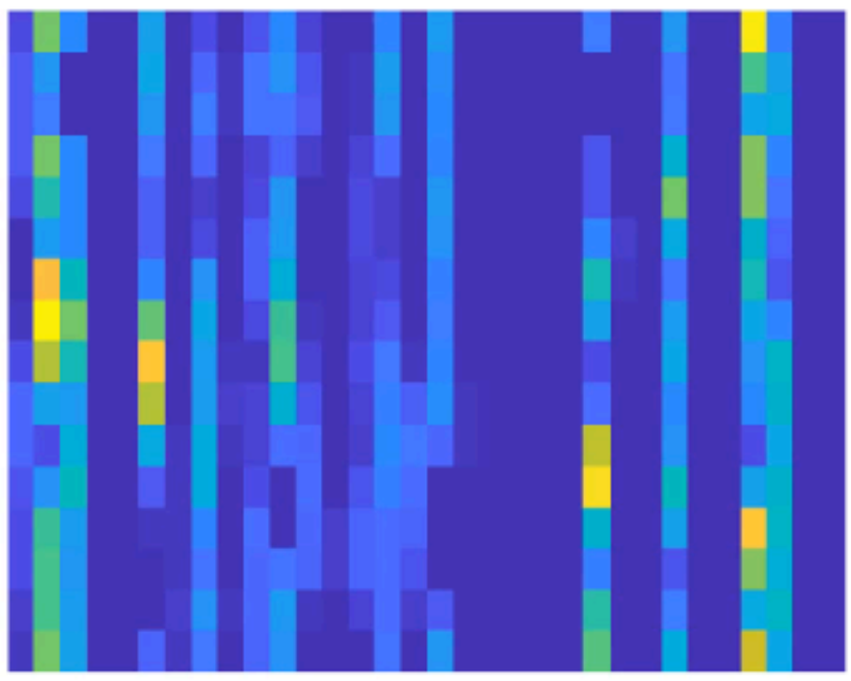




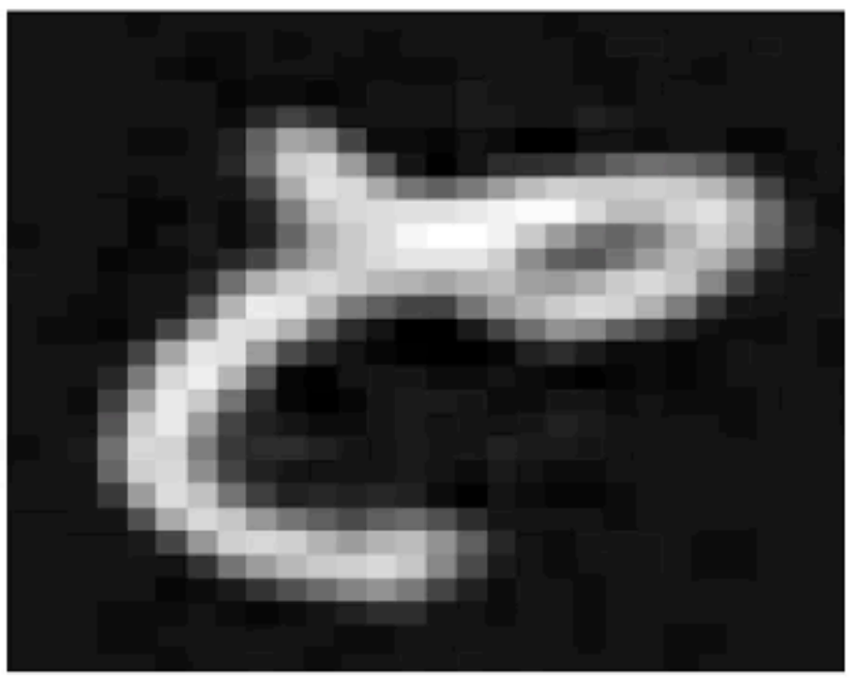
[1] Zhou et al ’16

Rotation-Equivariant Auto-encoder

RotDCF ConvAE
rc5x5x1x8 ReLu ap2x2
rc5x5xN_θx8x16 ReLu ap2x2
rc5x5xN_θx16x32 ReLu ap2x2
rc5x5xN_θx32x32 ReLu ← **Encoded representation**
fc128 ReLu ct5x5x128x16N_θ ReLu
ct5x5x16N_θx8N_θ (upsample 2x2) ReLu
ct5x5x8N_θx1 (upsample 2x2) Eucledian-loss



Code



Reconstruction

Outline

- Background
  - Convolutional neural network (CNN)
  - Network with pre-fixed weights
  - Decomposed Convolutional Filters
    - DCF Net
    - Stability analysis
    - Experiments
- RotDCF: Group-Equivariant DCF Net
- **Future Directions**

Future Directions

- Implementation
  - Level of parallelism in GPU
  - Memory efficiency
- Analysis
  - Completeness of the representation
  - Implication for generalization

Preprints

- X. Cheng, Q. Qiu, R. Calderbank, G. Sapiro. “Decomposed Filters in Convolutional Neural Networks”, in preparation.

  Short version: “DCFNet: Deep Neural Network with Decomposed Convolutional Filters", to appear at *ICML 2018*. [*arXiv: 1802.04145*]

- X. Cheng, Q. Qiu, R. Calderbank, G. Sapiro. “RotDCF: Decomposition of Convolutional Filters for Rotation-Equivariant Deep Networks”, [*arXiv: 1805.06846*]

Questions?

Thank You!