Bilinear Models for Fine-grained Visual Recognition

Tsung-Yu Lin Aruni RoyChowdhury Subhransu Maji

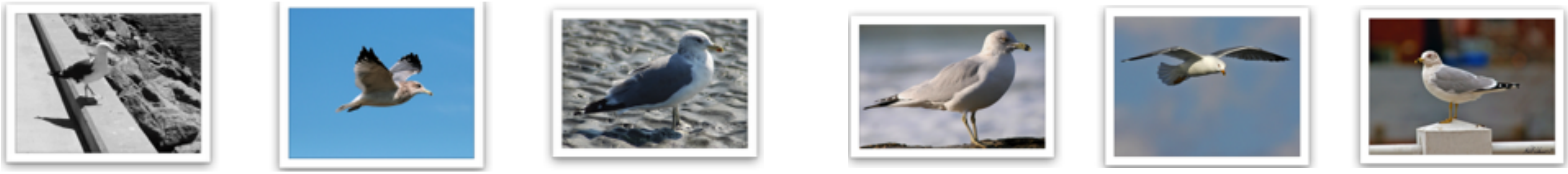
College of Information and Computer Sciences
University of Massachusetts, Amherst

Fine-grained visual recognition

- Example: distinguish between closely related categories

California gull

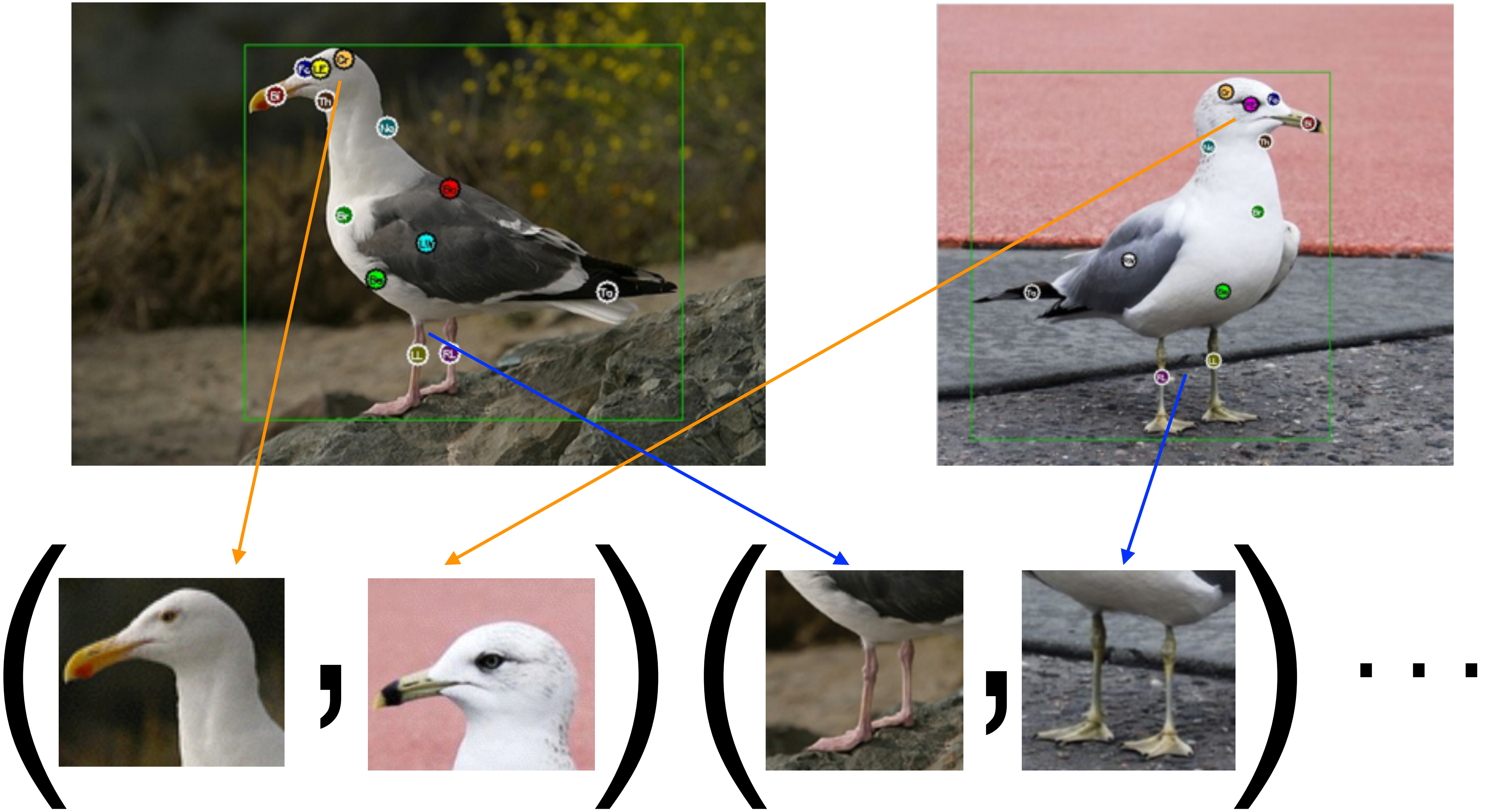
Ringed beak gull



- inter-category variation *v.s* intra-category variation
  - location, pose, viewpoint, background, lighting, gender, season, etc

Part-based models

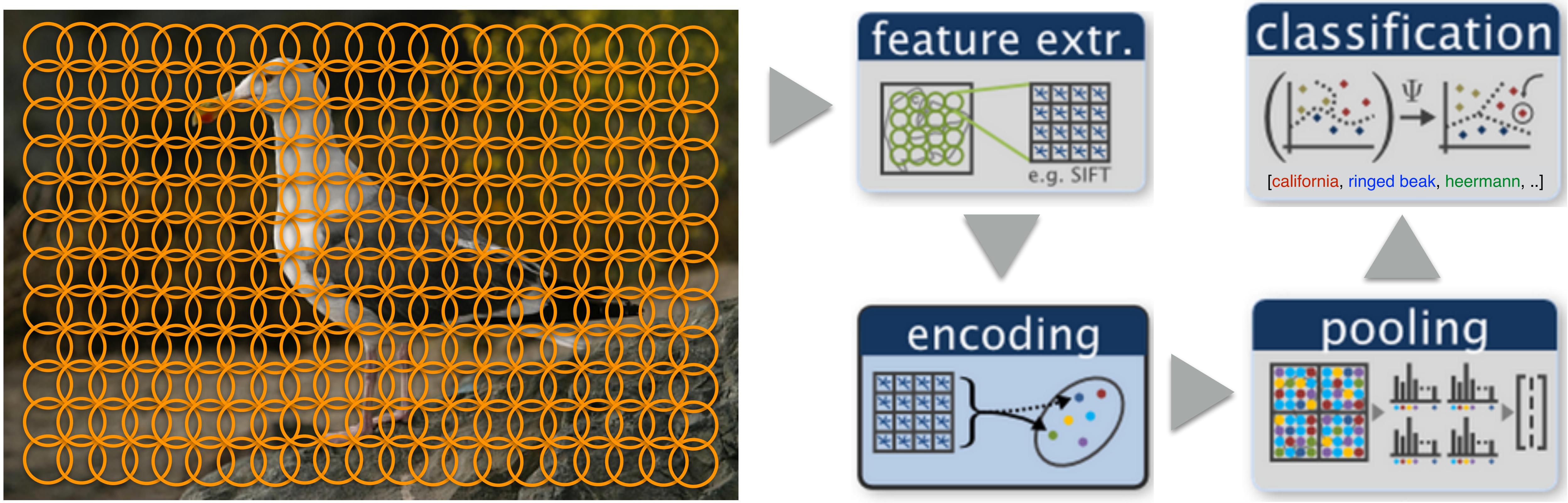
- Localize **parts** and compare corresponding locations



- Factor out the variation due to pose, viewpoint and location

General image classification

- **Classical approaches:** Image as a collection of patches
  - Orderless pooling and no explicit modelling of pose or viewpoint
  - Variants such as **Fisher vectors** work well for image classification



- **Modern approaches:** CNN, Fisher vector CNN [Cimpoi et al., CVPR15]

Tradeoffs

- Part-based models
  - ✓ Higher accuracy
  - x Part detection is slow
  - x Requires part annotations

- Examples:
  - Birdlets [Farrell et al.]
  - Part-based RCNN [Zhang et al.]
  - Pose-normalized CNNs [Branson et al.]

- Image classification models
  - ✓ Only requires image label
  - ✓ Faster evaluation
  - x Lower accuracy

- Examples:
  - Bag-of-visual-words [Csurka et al.]
  - Fisher vector [Jégou et al.]
  - VLAD [Perronnin et al.]
  - CNNs [Krizhevsky et al., ….]

- We propose bilinear models
  - Generalizes both **part-based** and **bag-of-visual-words** models
  - **Better accuracy** than part-based models w/o part annotations
  - Allows **fine-tuning** of features for bag-of-visual-words models

Bilinear models for classification

- A bilinear model for classification is a four-tuple

$$\mathcal{B} = (f_A, f_B, \mathcal{P}, \mathcal{C})$$

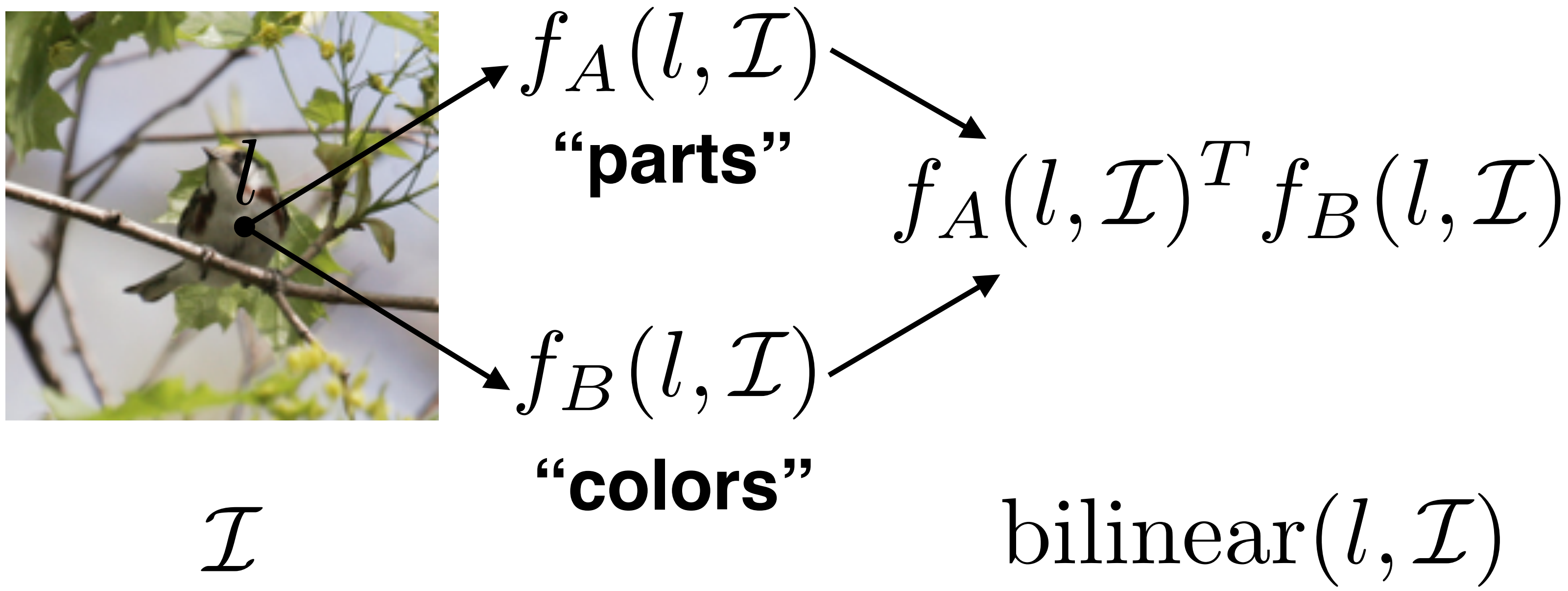
feature extractor

pooling

classification

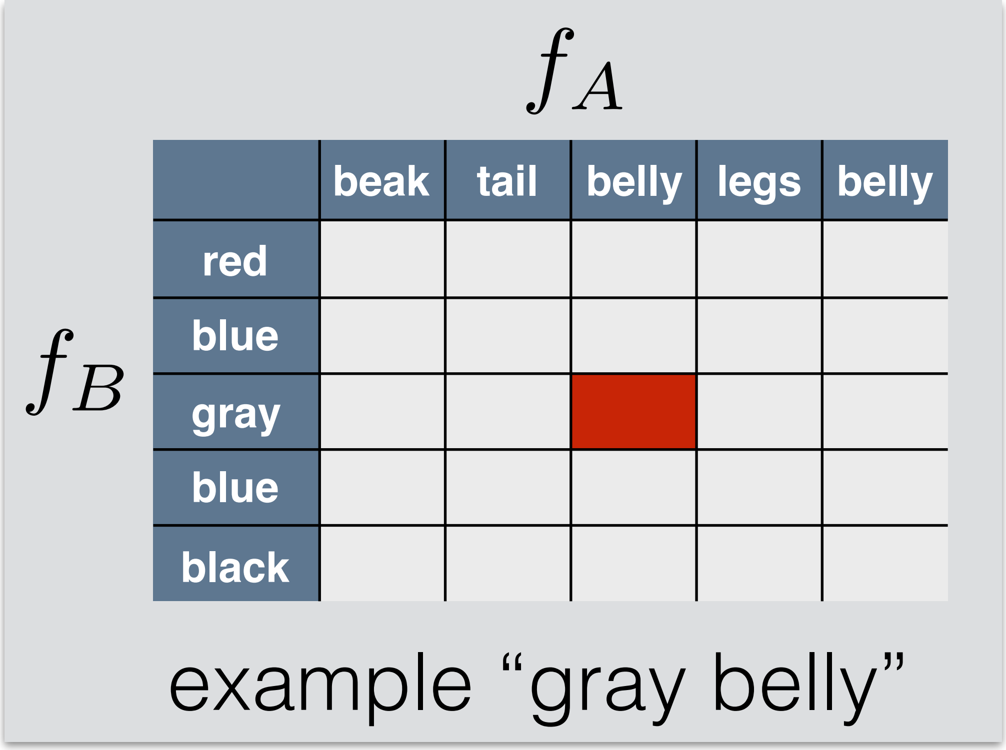
$f : \mathcal{L} \times \mathcal{I} \rightarrow R^{c \times D}$ *e.g.*, SIFT is $R^{1\text{x}128}$

image → local features



$$f_A(l, \mathcal{I})^T f_B(l, \mathcal{I})$$

$$\text{bilinear}(l, \mathcal{I})$$



f_A

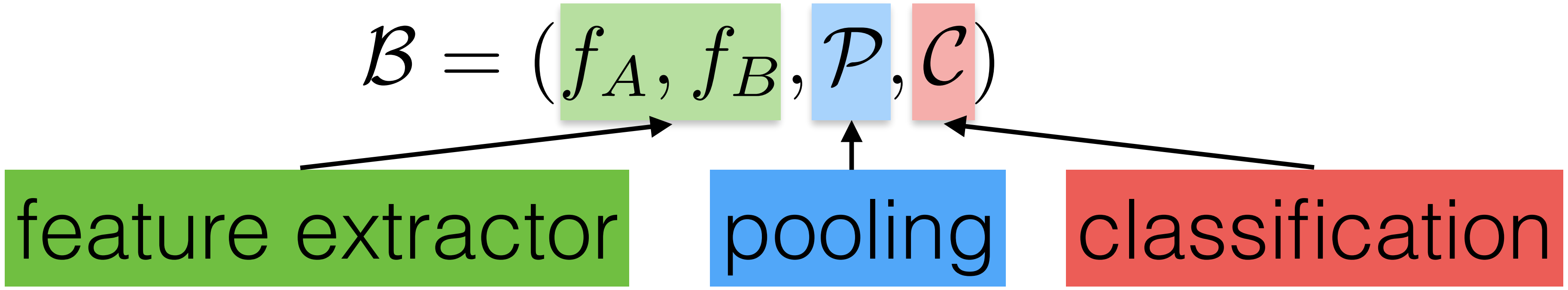
f_B

	beak	tail	belly	legs	belly
red					
blue					
gray			🟥		
blue					
black					

example “gray belly”

Bilinear models for classification

- A bilinear model for classification is a four-tuple



$$\mathcal{B} = (f_A, f_B, \mathcal{P}, \mathcal{C})$$

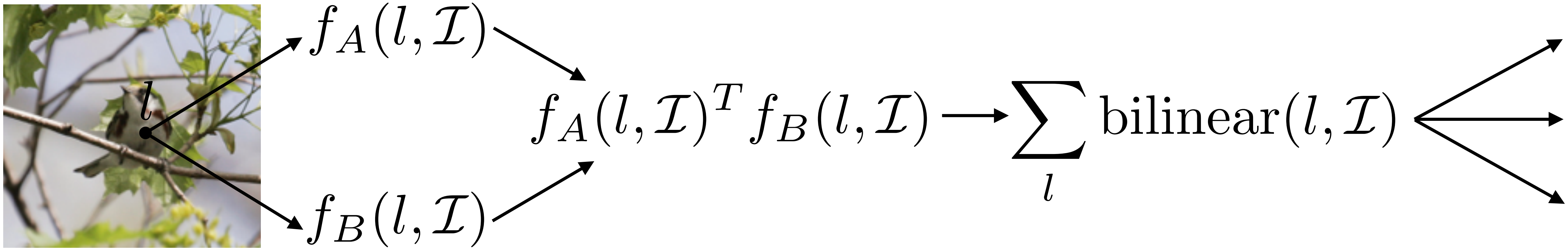
feature extractor

pooling

classification

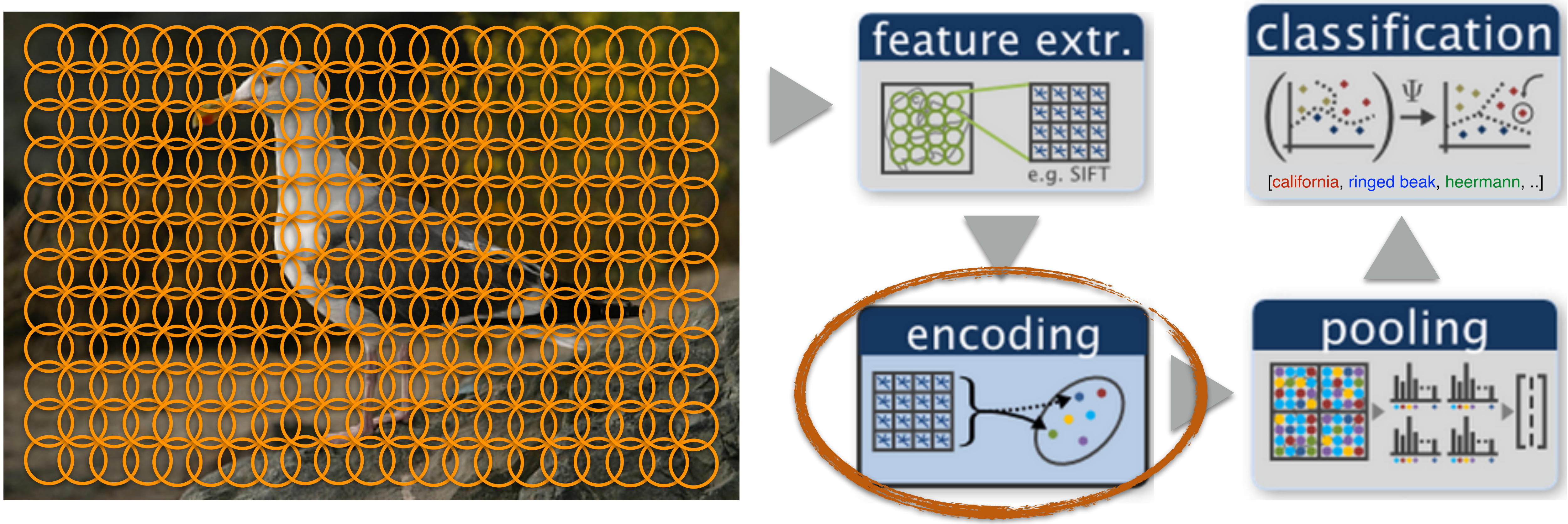
$$f : \mathcal{L} \times \mathcal{I} \rightarrow R^{c \times D}$$

image → local features → descriptor → class



$\mathcal{I}$ $\quad$ $\text{bilinear}(l, \mathcal{I})$ $\quad$ $\Phi(\mathcal{I})$

Fisher vector is a bilinear model



- **Fisher vector (FV)** models [Perronnin et al., 10]
  - Locally encode statistics of feature **x** weighted by η(**x**)

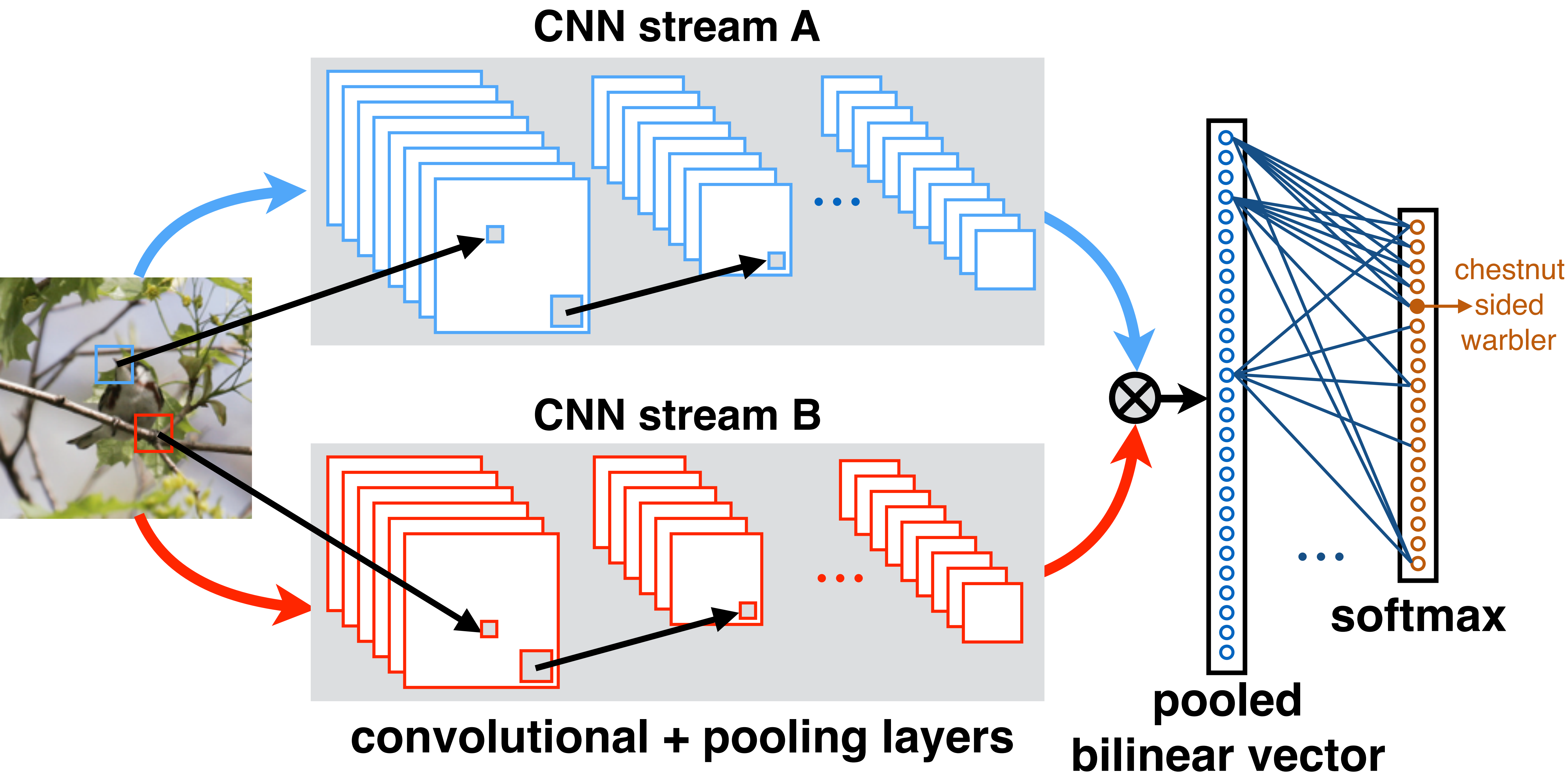
$$\alpha_i = \Sigma_i^{-\frac{1}{2}}(\mathbf{x} - \mu_i) \qquad \beta_i = \Sigma_i^{-1}(\mathbf{x}-\mu_i)\odot(\mathbf{x}-\mu_i)-1$$

- **FV** is bilinear model with

$$\begin{aligned} f_A &= [\alpha_1\ \beta_1; \alpha_2\ \beta_2; \ldots; \alpha_k\ \beta_k] \\ f_B &= \mathrm{diag}(\eta(\mathbf{x})) \end{aligned}$$

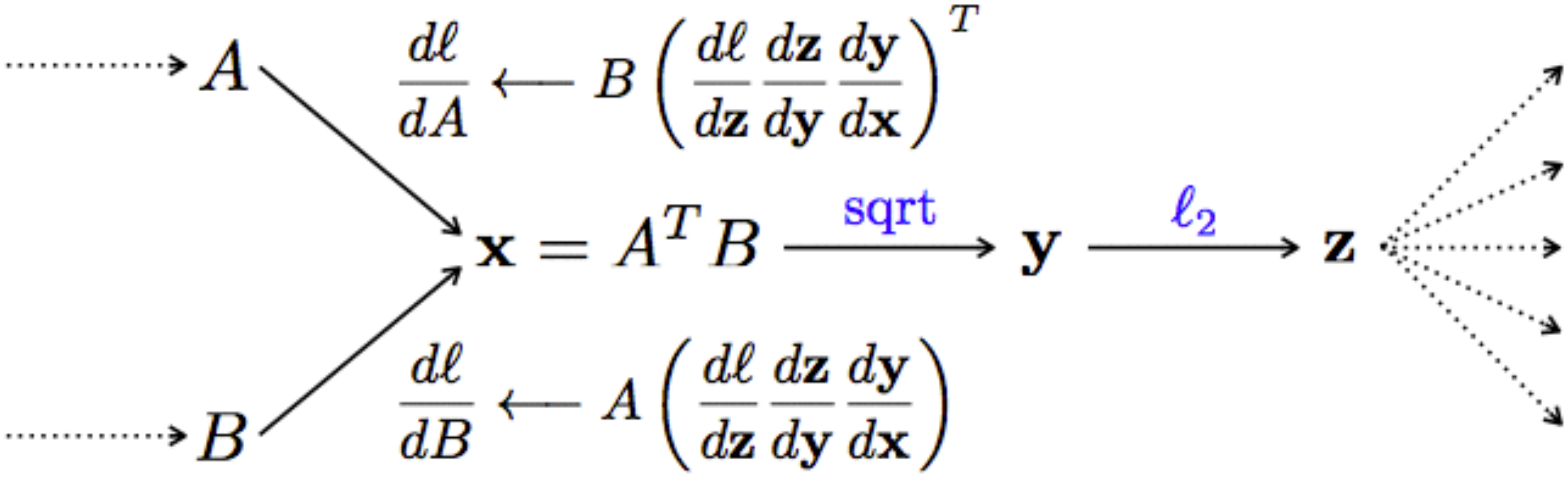
Bilinear CNN model

- Decouple f_A and f_B by using separate CNNs



Bilinear CNN model

- Back-propagation though the bilinear layer is easy



$$\frac{d\ell}{dA} \longleftarrow B\left(\frac{d\ell}{d\mathbf{z}}\frac{d\mathbf{z}}{d\mathbf{y}}\frac{d\mathbf{y}}{d\mathbf{x}}\right)^T$$

$$\mathbf{x} = A^T B \xrightarrow{\text{sqrt}} \mathbf{y} \xrightarrow{\ell_2} \mathbf{z}$$

$$\frac{d\ell}{dB} \longleftarrow A\left(\frac{d\ell}{d\mathbf{z}}\frac{d\mathbf{z}}{d\mathbf{y}}\frac{d\mathbf{y}}{d\mathbf{x}}\right)$$

- Allows **end-to-end** training

Experiments: Methods

- **Local features:**
  - SIFT descriptor [Lowe ICCV99]
  - VGG-M (**5 conv** + 2 fc layers) [Chatfield et al., BMVC14]
  - VGG-VD (**16 conv** + 2 fc layers) [Simonyan and Zisserman, ICLR15]

- **Pooling architectures:**
  - Fully connected pooling **(FC)**
  - Fisher vector pooling **(FV)**
  - Bilinear pooling **(B)**

- **Notation examples:**
  - **FC-CNN (M)** — Fully connected pooling with VGG-M
  - **FV-CNN (D)** — Fisher vector pooling with VGG-VD [Cimpoi et al.,15]
  - **B-CNN (D, M)** — Bilinear pooling with VGG-D and VGG-M

Experiments: Datasets

small, clutter

CUB 200-2011
200 species
11,788 images

FGVC Aircraft
100 variants
10,000 images

clutter

Stanford cars
196 models
16,185 images

- All models are trained with image labels only
  - No part or object annotations are used at training or test time

Results: Birds classification

- Accuracy on CUB 200-2011 dataset
- Setting: provided with only the image at test time

Method	**w/o ft**	**w/ ft**
FV-SIFT	18.8	

Results: Birds classification

- Accuracy on CUB 200-2011 dataset
- Setting: provided with only the image at test time

Method	**w/o ft**	**w/ ft**
FV-SIFT	18.8	
FC-CNN (M)	52.7	

Results: Birds classification

- Accuracy on CUB 200-2011 dataset
- Setting: provided with only the image at test time

Method	**w/o ft**	**w/ ft**
FV-SIFT	18.8	
FC-CNN (M)	52.7	
FV-CNN (M)	61.1	

Results: Birds classification

- Accuracy on CUB 200-2011 dataset
- Setting: provided with only the image at test time

Method	**w/o ft**	**w/ ft**
FV-SIFT	18.8	
FC-CNN (M)	52.7	
FV-CNN (M)	61.1	
B-CNN (M,M)	72.0	

Results: Birds classification

- Accuracy on CUB 200-2011 dataset
- Setting: provided with only the image at test time

Method	**w/o ft**	**w/ ft**
FV-SIFT	18.8	-
FC-CNN (M)	52.7 →	58.8
FV-CNN (M)	61.1	
B-CNN (M,M)	72.0	

fine-tuning helps

Results: Birds classification

- Accuracy on CUB 200-2011 dataset
- Setting: provided with only the image at test time

Method	**w/o ft**	**w/ ft**
FV-SIFT	18.8	-
FC-CNN (M)	52.7	58.8
FV-CNN (M)	61.1	64.1
B-CNN (M,M)	72.0	

direct fine-tuning is hard so use ft FC-CNN models

indirect fine-tuning helps

outperforms multi-scale FV-CNN Cimpoi et al. CVPR 15

Results: Birds classification

- Accuracy on CUB 200-2011 dataset
- Setting: provided with only the image at test time

Method	**w/o ft**	**w/ ft**
FV-SIFT	18.8	-
FC-CNN (M)	52.7	58.8
FV-CNN (M)	61.1	64.1
B-CNN (M,M)	72.0	78.1

direct fine-tuning is hard so use ft FC-CNN models

indirect fine-tuning helps

outperforms multi-scale FV-CNN Cimpoi et al. CVPR 15

Results: Birds classification

- Accuracy on CUB 200-2011 dataset
- Setting: provided with only the image at test time

Method	w/o ft	w/ ft
FV-SIFT	18.8	-
FC-CNN (M)	52.7	58.8
FC-CNN (D)	61.0	70.4
FV-CNN (M)	61.1	64.1
B-CNN (M,M)	72	78.1

Results: Birds classification

- Accuracy on CUB 200-2011 dataset
- Setting: provided with only the image at test time

Method	w/o ft	w/ ft
FV-SIFT	18.8	-
FC-CNN (M)	52.7	58.8
FC-CNN (D)	61.0	70.4
FV-CNN (M)	61.1	64.1
FV-CNN (D)	71.3	74.7
B-CNN (M,M)	72	78.1

Results: Birds classification

- Accuracy on CUB 200-2011 dataset
- Setting: provided with only the image at test time

Method	w/o ft	w/ ft
FV-SIFT	18.8	-
FC-CNN (M)	52.7	58.8
FC-CNN (D)	61.0	70.4
FV-CNN (M)	61.1	64.1
FV-CNN (D)	71.3	74.7
B-CNN (M,M)	72	78.1
B-CNN (D,M)	80.1	**84.1**
B-CNN (D,D)	80.1	84.0

Results: Birds classification

- Accuracy on CUB 200-2011 dataset
- Setting: provided with only the image at test time

Method	w/o ft	w/ ft
FV-SIFT	18.8	-
FC-CNN (M)	52.7	58.8
FC-CNN (D)	61.0	70.4
FV-CNN (M)	61.1	64.1
FV-CNN (D)	71.3	74.7
B-CNN (M,M)	72	78.1
B-CNN (D,M)	80.1	**84.1**
B-CNN (D,D)	80.1	84.0
SoTA	**84.1** [1], 82.0 [2], 73.9 [3], 75.7 [4]	

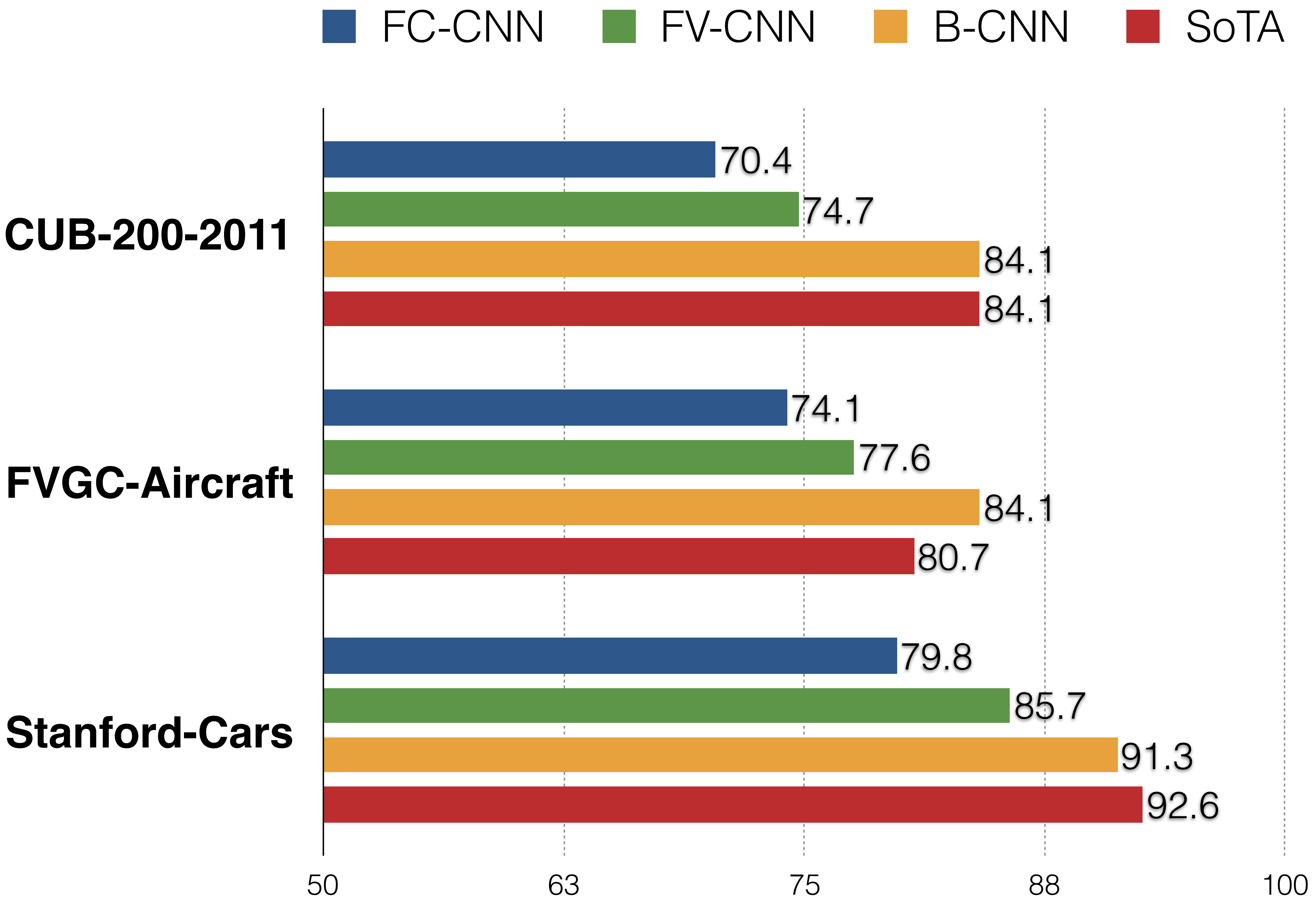
[1] Spatial Transformer Networks, Jaderberg et al., NIPS 15

[2] Fine-Grained Rec. w/o Part Annotations, Krause et al., CVPR 15 (+ object bounding-boxes)

[3] Part-based R-CNNs, Zhang et al., ECCV 14 (+ part bounding-boxes)

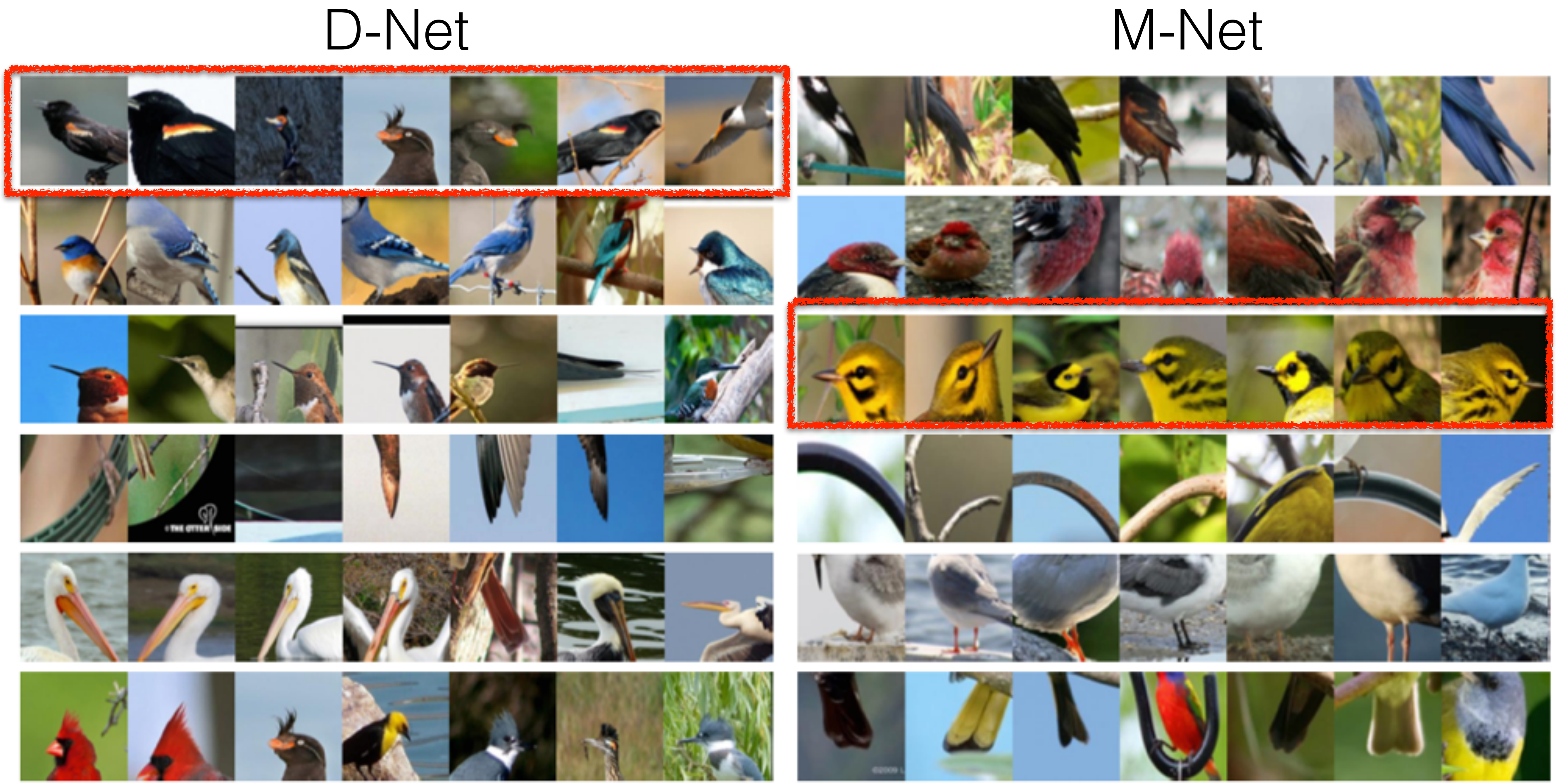
[4] Pose normalized CNNs, Branson et al., BMVC 14 (+ landmarks)

Results: Comparison



Model visualization

- Visualizing top activation on B-CNN(D,M)

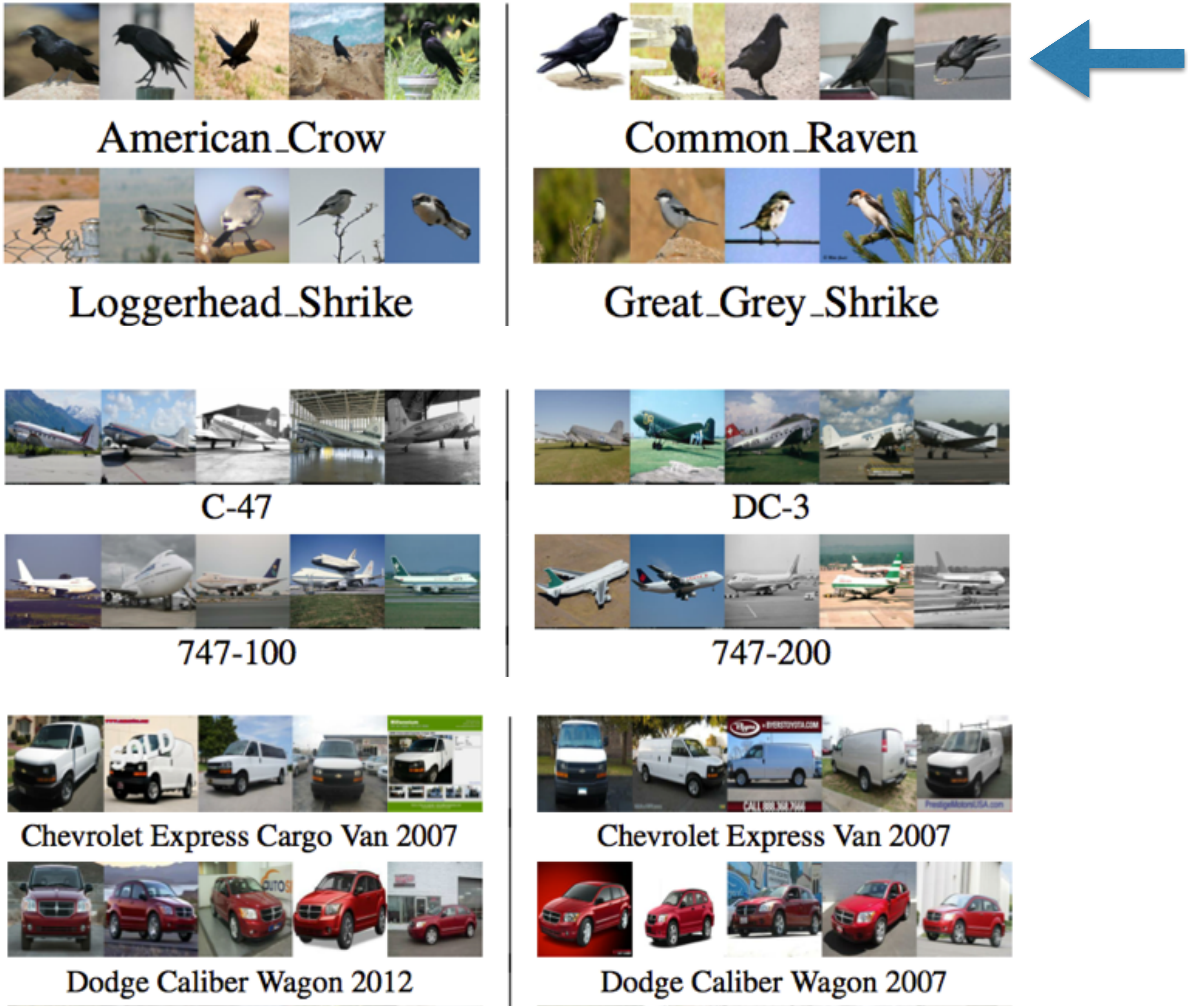


Most confused categories

CUB-200

Aircrafts

Stanford cars



Conclusion

- Bilinear models
    - generalize both **part-based** and **bag-of-visual-words models**
    - achieve **high** accuracy on fine-grained recognition tasks **without** additional annotations

- Fast at test time
    - B-CNN [D, D] runs at **10 images/second** on TeslaK40 GPU

- Code and pre-trained models available
    - more details here: http://vis-www.cs.umass.edu/bcnn

- Come by our poster [#68] for more details