Bilinear CNN Models for Fine-grained Visual Recognition

Tsung-Yu Lin Aruni RoyChowdhury Subhransu Maji

University of Massachusetts, Amherst

http://vis-www.cs.umass.edu/bcnn

Code Available

Task: Fine-grained recognition

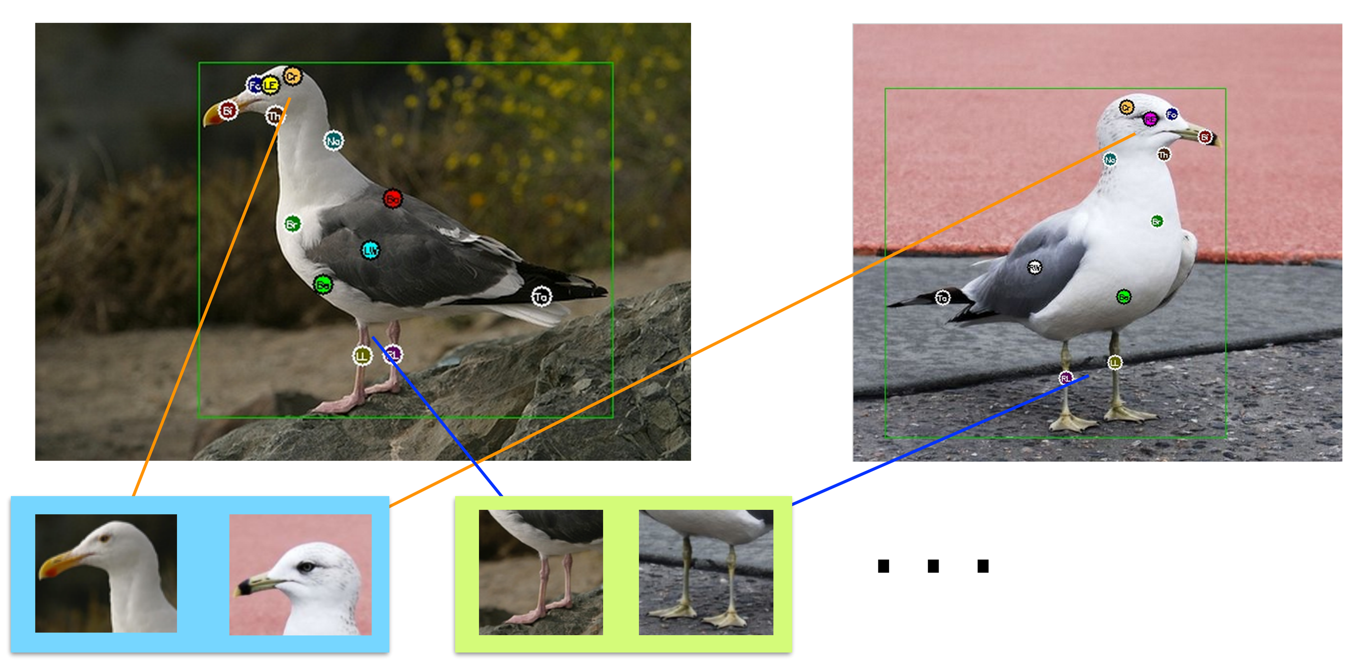
- **Example**: distinguish the two bird species

California gull Ringed beak gull

- **Challenge**: intra-category variation v.s. inter-category variation
  - Location, pose, viewpoint, background, lighting, gender, etc

Approach 1: Part-based models

- Localize **parts** and compare corresponding locations

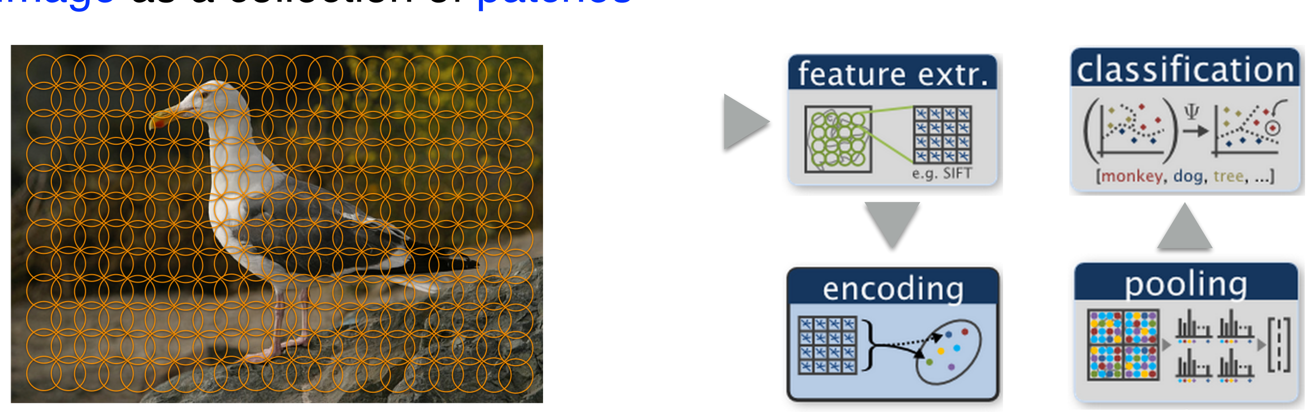


✓ Accuracy is high
✗ Part detection is slow
✗ Requires part annotation

Recent examples:
- Part-based R-CNN [2]
- Pose normalized CNNs [3]

Approach 2: Texture-based models

- Image as a collection of patches



✓ Can be trained w/ image labels
✓ Fast CPU implementations
✗ Lower accuracy

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

Goal: combine the best of both approaches

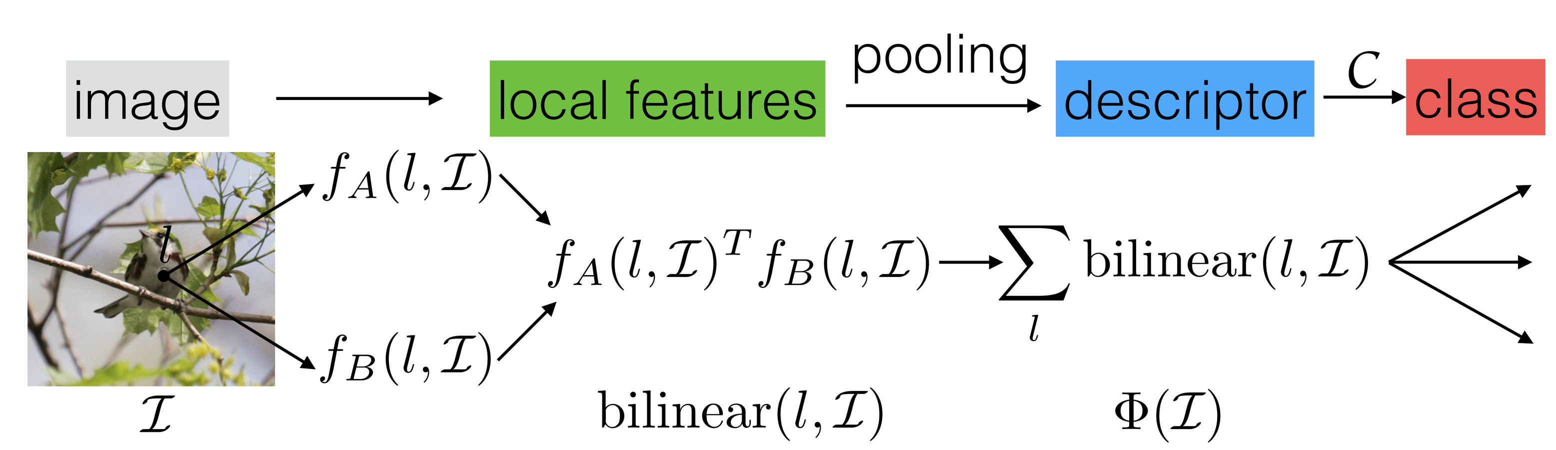
Proposed approach: Bilinear CNN model

- **Bilinear model** is a four tuple:

$f : \mathcal{L} \times \mathcal{I} \rightarrow R^{c \times D}$ $\mathcal{B} = (f_A, f_B, \mathcal{P}, \mathcal{C})$

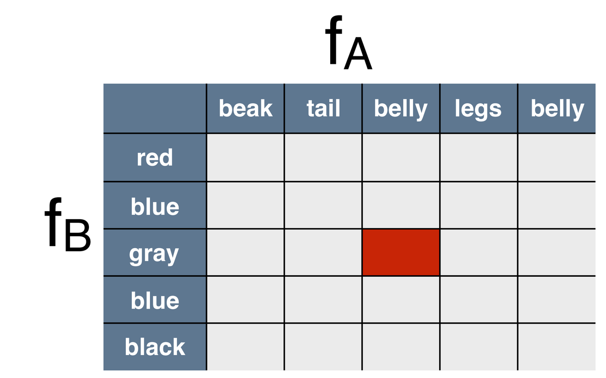
feature extractor pooling classification

- **Classification pipeline:**
  1. For each location l, extract features $f_A(l, \mathcal{I})$ and $f_B(l, \mathcal{I})$
  2. Take the outer product: $\text{bilinear}(l, \mathcal{I}) = f_A(l, \mathcal{I})^T f_B(l, \mathcal{I})$
  3. Pool across locations: $\Phi(\mathcal{I}) = \sum_l \text{bilinear}(l, \mathcal{I})$
  4. Predict class probability: $\text{softmax}(\mathbf{w}^T \Phi(\mathcal{I}))$



- **Motivation:**
  - Model pairwise feature interactions in a translationally invariant manner
  - Compositional features — O(n²) representation with O(n) features
  - End-to-end learning of parameters

- **Bilinear models** generalize both **part-based** models and **texture** models
  - If f_A is a part detector and f_B is a color descriptor then the outer product captures all part-color interactions (left)
  - Classical texture descriptors such as the Fisher vector can be written as **bilinear model** with a suitable choice of f_A and f_B (right)

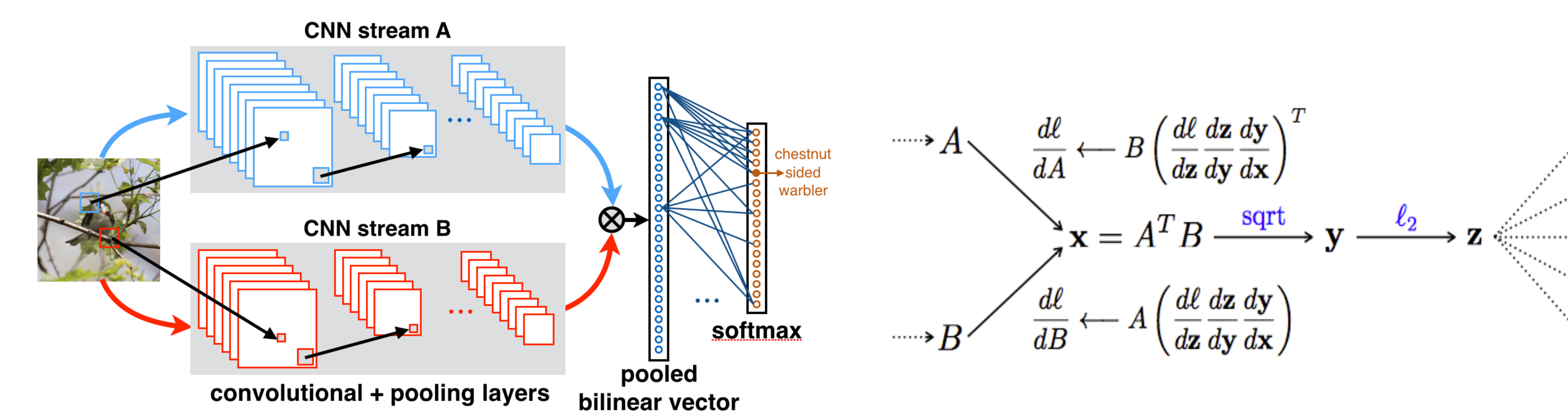


example "gray belly"

$$\alpha_i = \Sigma_i^{-\frac{1}{2}}(\mathbf{x} - \mu_i)$$
$$\beta_i = \Sigma_i^{-1}(\mathbf{x} - \mu_i) \odot (\mathbf{x} - \mu_i) - 1$$
$$f_A = [\alpha_1\ \beta_1; \alpha_2\ \beta_2; \ldots; \alpha_k\ \beta_k]$$
$$f_B = \text{diag}(\eta(\mathbf{x}))$$

Fisher vector as a bilinear model

- **Training:** Extract f_A and f_B using separate CNNs



- Back-propagation though the bilinear layer is easy
- Allows end-to-end training with just using image labels (unlike Fisher vectors)
- Two normalization layers, square-root and l_2 normalization, improve results

Experiments

- **Classification accuracy:**
  - Using image labels only (**no** part or bounding-box annotations)
  - CNNs used: VGG-M [M] (Chatfield et al.) and VGG-VERYDEEP-16 [D] (Simonyan et al.)

	Method	CUB-200-2011 w/o ft	CUB-200-2011 w/ ft	FGVC-Aircraft w/o ft	FGVC-Aircraft w/ ft	Stanford Cars w/o ft	Stanford Cars w/ ft
Fisher vector SIFT	FV-SIFT	18.8	-	61.0	-	59.2	-
Fully connected CNN (standard CNN w/ fc layer)	FC-CNN [M]	52.7	58.8	44.4	57.3	37.3	58.6
	FC-CNN [D]	61.0	70.4	45.0	74.1	36.5	79.8
Fisher vector CNN (Cimpoi et al., CVPR 15)	FV-CNN [M]	61.1	64.1	64.3	70.1	70.8	77.2
	FV-CNN [D]	71.3	74.7	70.4	77.6	75.2	85.7
Bilinear CNN (proposed approach)	B-CNN [M,M]	72.0	78.1	72.7	77.9	77.8	86.5
	B-CNN [D,M]	80.1	**84.1**	78.4	83.9	83.9	**91.3**
	B-CNN [D,D]	80.1	84.0	76.8	**84.1**	82.9	90.6
	Previous Work	**84.1** [1], 73.9 [2], 75.7 [3]		80.7 [5]		**92.6** [4], 82.7 [5]	

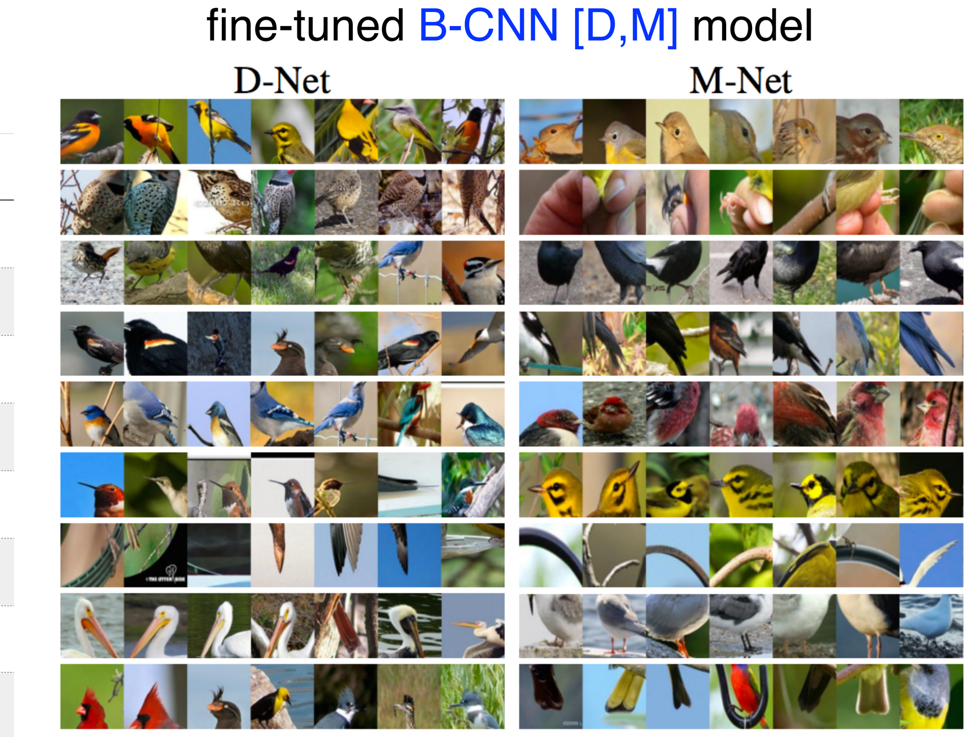
- **Effect of fine-tuning:**
  - FC-CNN: big improvements on aircrafts (**29%**) and cars (**43%**)
  - FV-CNN: Indirect fine-tuning, i.e. using fine-tuned FC-CNN for FV-CNN, leads to **3-10%** improvement
  - B-CNN: Fine-tuning improves results (**4-7%**)
- Fairly efficient during testing: **10 fps** for the **B-CNN [D,D]** model
- Translational invariance: **84.1%** (w/o b-box) vs **85.1%** (w/ b-box)

References

[1] Spatial transformer networks, Jaderberg et al. NIPS'15.
[2] Part-based R-CNN for fine-grained category detection, Zhang et al. ECCV'14
[3] Bird species categorization using pose normalized deep convolutional nets, Branson et al., BMVC'14.
[4] Fine-grained recognition w/o part annotations, Krause et al., CVPR'15
[5] Revisiting the fisher vector for fine-grained classification, Gosselin et al., Pattern Recognition Letters, 2014.
[6] Deep filter-banks for texture recognition and segmentation, Cimpoi et al., CVPR'14

- **Visualizations on the CUB dataset**

Top activations of various conv5 filters in the fine-tuned B-CNN [D,M] model



Most confused class pairs