Bilinear Models for Fine-grained Visual Recognition

Akilesh B, CS13B1042

Indian Institute of Technology, Hyderabad

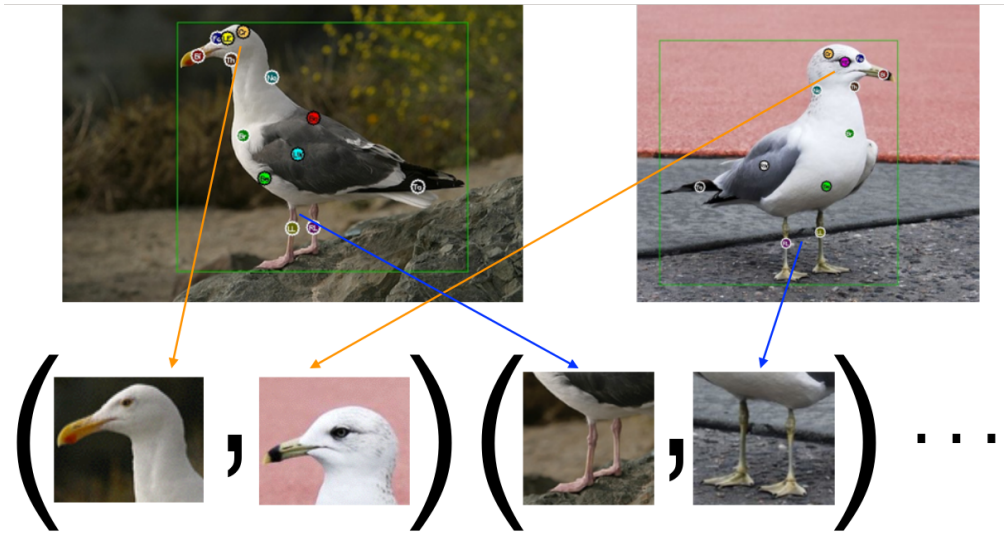
April 12, 2016

Introduction

- Fine-grained recognition tasks such as identifying the species of a bird, or the model of an aircraft, are quite challenging.
- The visual differences between the categories are small and can be easily overwhelmed by those caused by factors such as pose, viewpoint, or location of the object in the image.
- For example, the inter-category variation between Ringed-beak gull and a California gull due to the differences in the pattern on their beaks is significantly smaller than the inter-category variation on a popular fine-grained recognition dataset for birds.

Related Work

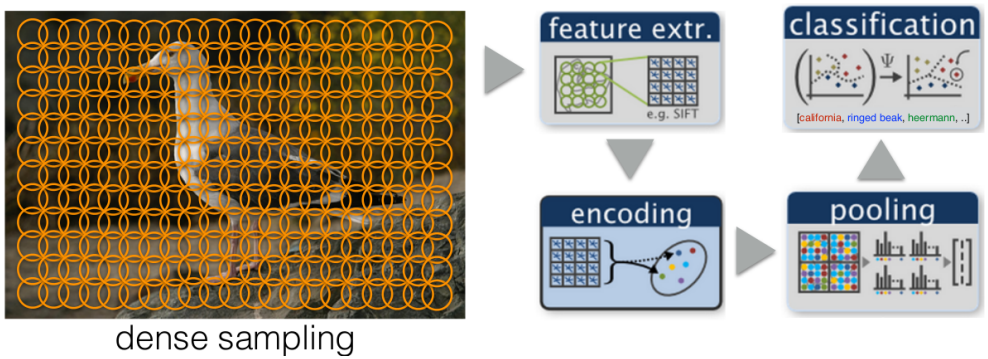
Figure: Part-based model



- Localize various parts of the object and model the appearance conditioned on their detected locations.
- The parts are manually defined and part detectors are trained in a supervised manner.
- It factors out variation due to pose, viewpoint and location.

Related Work

Figure: Texture models



- Image as a collection of patches [bag-of-visual words, Csurka et al 04].
- Orderless pooling and no explicit modeling of pose or viewpoint.
- Invariances due to: choice of features (eg. SIFT is robust to lighting changes). There is encoding + pooling + classification.
- Eg. Fisher Vectors work remarkably well for fine-grained tasks.

Tradeoffs

- Part-based models - Have the best recognition accuracy on many fine-grained recognition datasets (eg. birds).
- It is relatively slow since it involves part detection.
- Needs part annotations for training. This can be time consuming and may require expert knowledge. Parts may be hard to define them for some categories.
- Texture models - Easy to deploy since they only need image labels for training. They have very fast CPU implementations.
- However, they have lower recognition accuracy. The pipelined procedure (features $\rightarrow$ encoding $\rightarrow$ classification) can be sub-optimal. For instance, the feature extractors are not learned.
- Can we get the best of both?

Bilinear models for classification

Figure: Bilinear models

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

feature extractor | pooling | classification

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

image → local features

$f_A(l, \mathcal{I})$ "parts"

$f_B(l, \mathcal{I})$ "colors"

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

$\mathcal{I}$

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

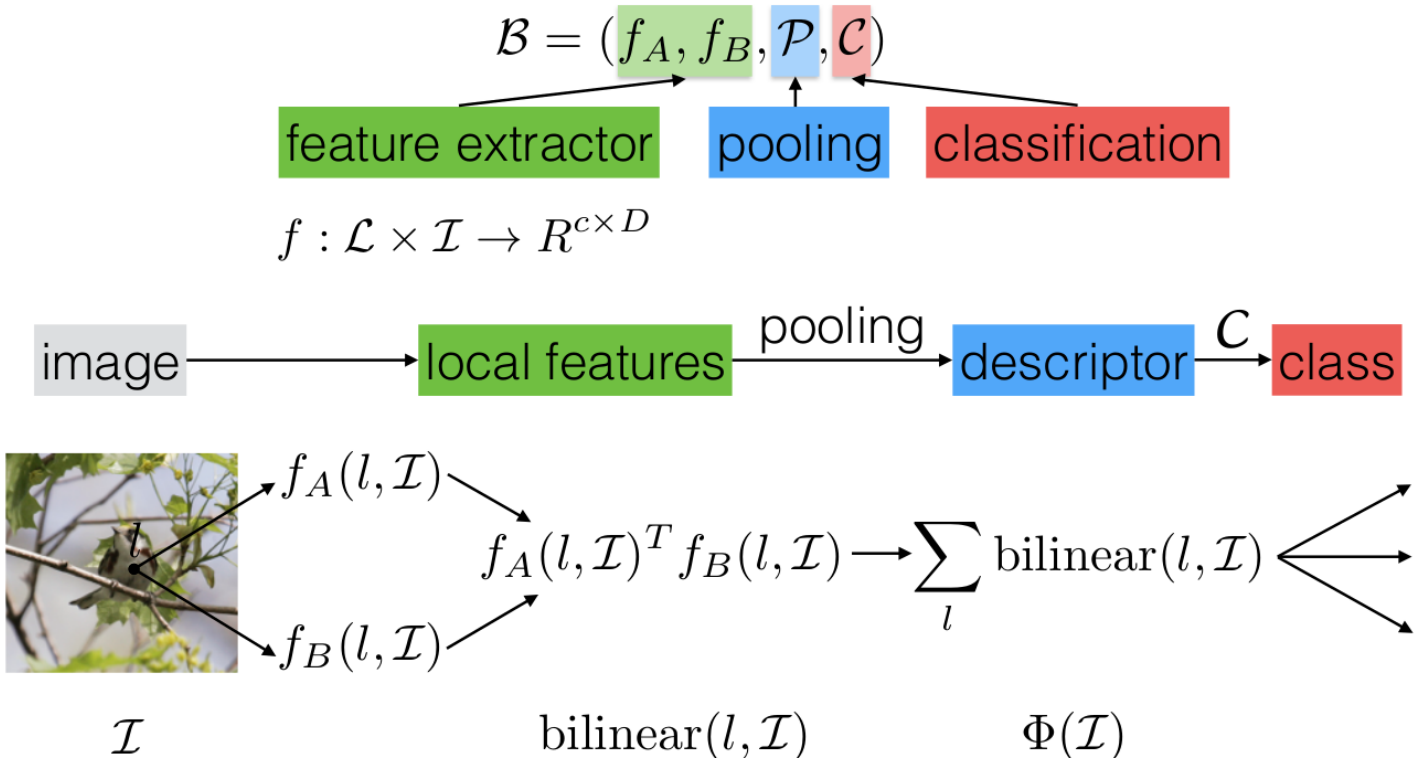
f_A

f_B	beak	tail	belly	legs	belly
red					
blue					
gray			■		
blue					
black					

example "gray belly"

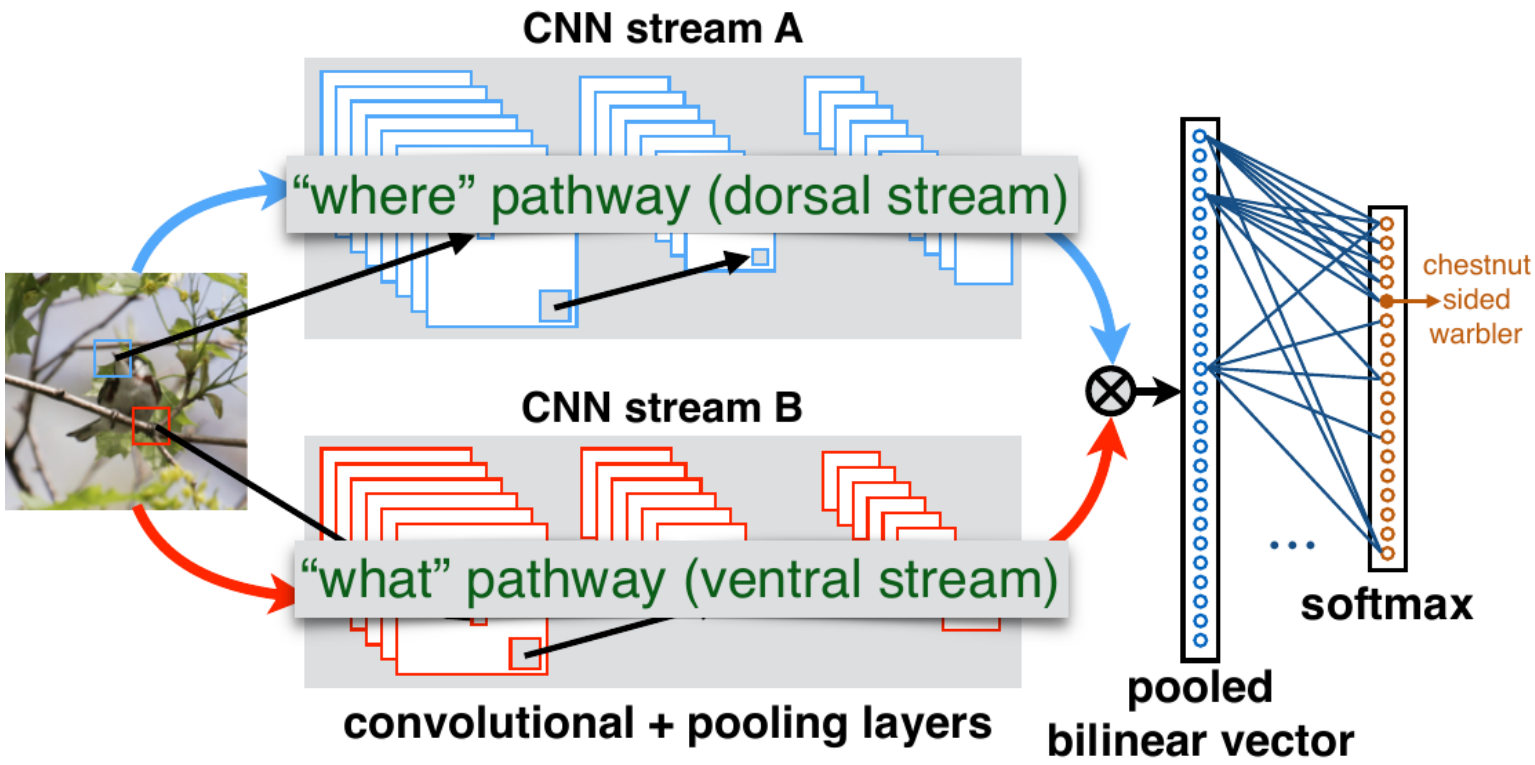
Bilinear models for classification

Figure: Bilinear models



Bilinear CNN model

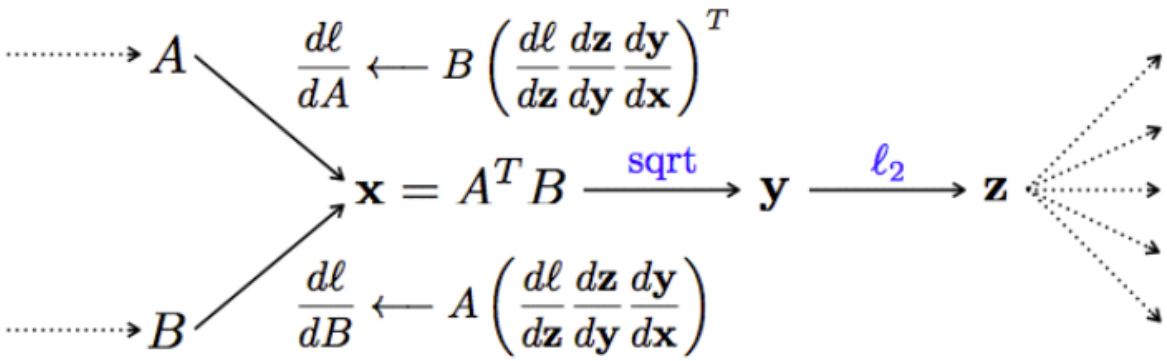
Figure: Bilinear CNN model



- Generalization: decouple f_A and f_B by using separate feature functions

Bilinear CNN model training

Figure: Bilinear CNN model training



- Back-propagation though the bilinear layer is easy.
- It allows end-to-end training.

There are two normalization layers in the architecture

- Square-root normalization

$$y = sign(x)\sqrt{x} \tag{1}$$

- l_2 normalization

$$z = \frac{y}{\|y\|}$$

- Both these improve performance.

Experiments

Two CNN models initialized from ImageNet are considered:

- VGG-M (5 convolutional layers + 2 fully connected layers)
- VGG-D (13 convolutional layers + 2 fully connected layers)

Trained using image labels only (no part or bounding-box annotations).

Dataset

- CUB-200-2011 dataset. Birds+box uses bounding boxes at training and test time. It has 200 categories of birds, 11788 images. It contains 15 part locations, 312 binary attributes and 1 bounding box.

Classification demo

Figure: Test image

Figure: Classification Demo

```
>> bird_test_aki
0.11s to load imdb.
2.73s to load models into memory.
Top 5 prediction for test_image.jpg:
064.Ring_billed_Gull
059.California_Gull
147.Least_Tern
062.Herring_Gull
060.Glaucous_winged_Gull
2.21s to make predictions [GPU=0]
>>
```

Results: Birds classification

- Per-image accuracy on CUB 200-2011 dataset.
- Provided with only the image at test time.

Method	w/o ft	w/ ft
B-CNN [M,M]	71.6	77.4
B-CNN [M,D]	79.6	83.2
B-CNN [D,D]	79.5	82.8

Table: Fine-grained classification results

Extension to the current work

- In the current work, the authors propose bilinear CNN models for fine-grained visual recognition and apply it to birds, aircrafts and cars dataset.
- In their extension work, they propose one-to-many face recognition using Bilinear CNNs. They perform the experiments on Face-Scrub and IJB-A Train dataset.
- They use a linear SVM classifier learned for each person in gallery and max-pooling features or classifier scores to aggregate multiple media.
- Finally, they demonstrate how a standard CNN pre-trained on a large face database (say VGG-Face model) can be converted into a B-CNN without any additional feature training.

Most confused birds

Figure: Most confused birds

References

Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji: Bilinear CNNs for Fine-grained Visual Recognition, ICCV 2015

Aruni RoyChowdhury, Tsung-Yu Li, and Subhransu Maji: One to many face recognition using Bilinear CNNs, WACV 2016