Metric Learning For Clustering Images From Unknown Classes

Chinmayee Shah

Problem

Learn embeddings so that images from similar classes cluster together, even for classes not present during training.

- Then cluster with K-means over learned features.
- Clustering using hand-crafted features and distance metrics does not perform well.
- Instead, learn deep features using convolutional neural networks.

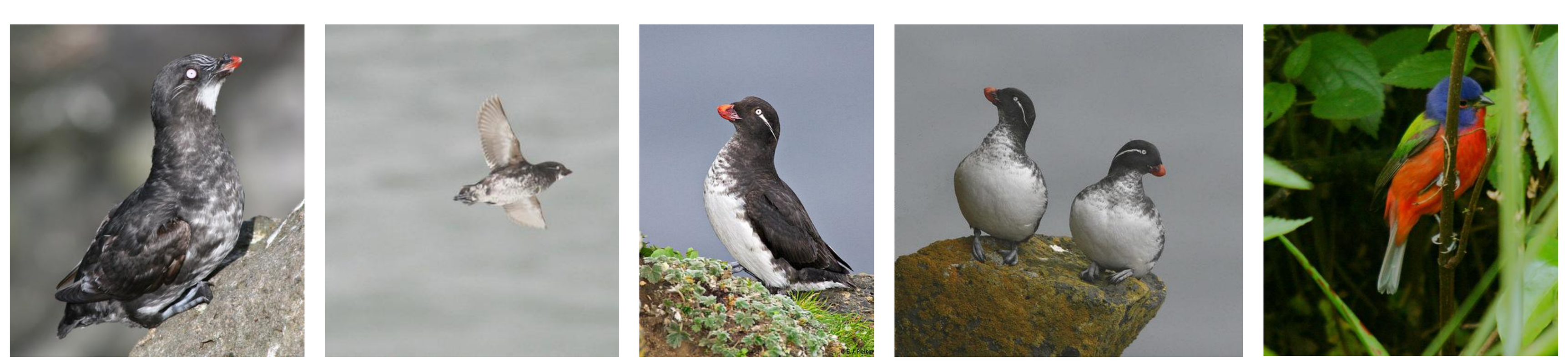
Question: How to model the loss, and train for classes not present at training time?

Comparison to Image Classification

- Training data for all classes is not available - unknown classes at test time
- Only a few images per class, and many classes
- Variance between images within a class may be high, compared to images between classes

Data-set

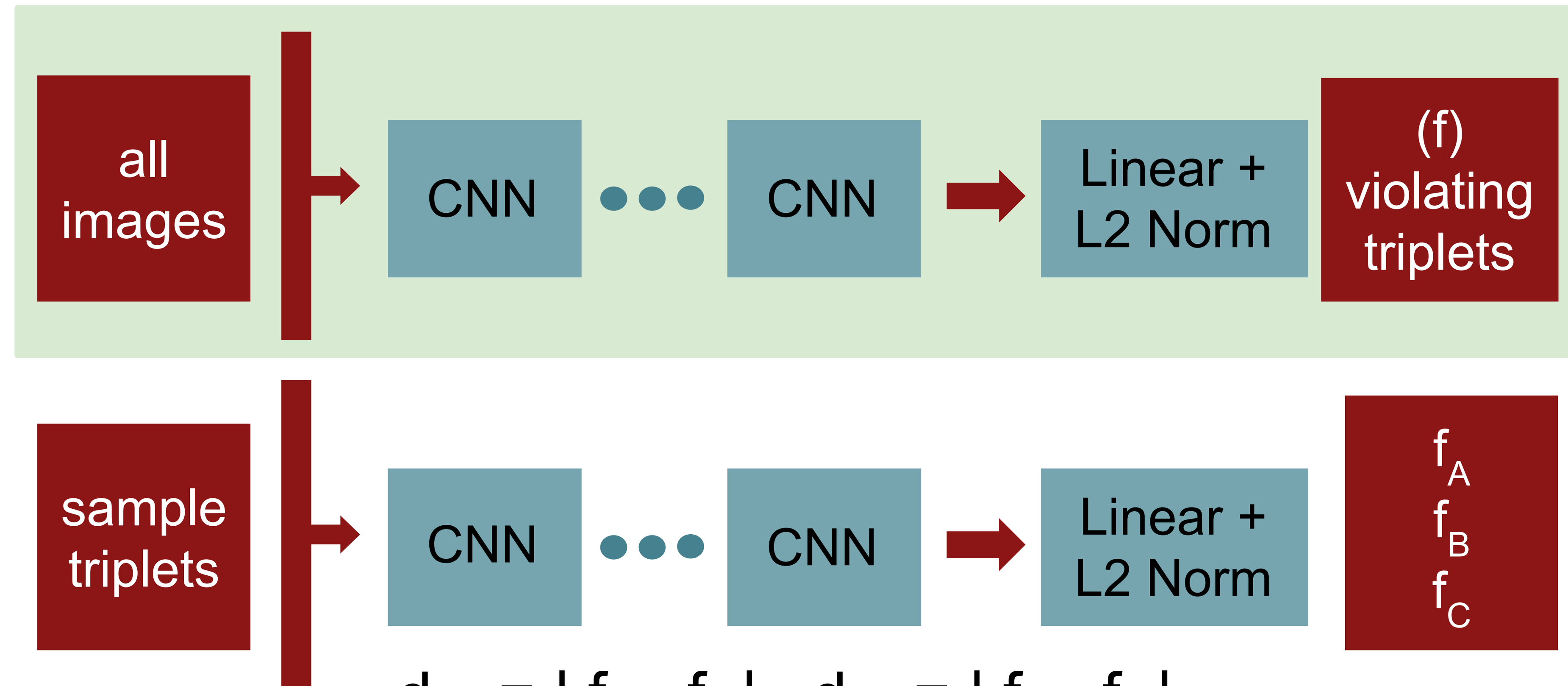
CUB-200-2011 birds: 200 species, 50 per specie



A A B B C

Model

Triplets: anchor (a), positive (p), negative(n)



$d_{ap} = | f_a - f_p |$, $d_{an} = | f_a - f_n |$

- $\mathbf{d_{ap} < d_{an}}$: positive closer to anchor than negative
- Cluster images using learned features f
- Hinge loss and ratio loss over distances
- Sampling strategies for triplets-at-input model
  - Hard: least d_{an} , Semi-hard: $d_{an} < d_{ap}$
  - Incorrectly clustered images

Networks and Training

Network Architectures

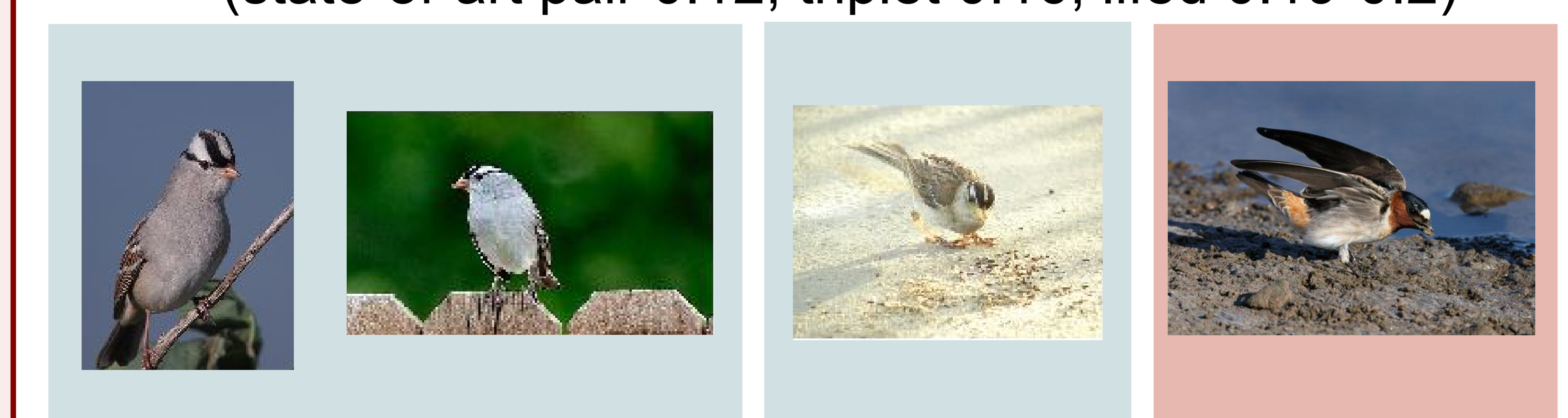
- Custom: train from scratch
- SqueezeNet, Inception3, Resnet50: use pretrained activation layers, optionally replace final classifier layer by a fully connected layer to learn features

Data Split

- 80 + 40 classes for training, 40 for validation/tuning
- 80 for testing

Results

- Using all triplets with non-zero loss far more stable than any sampling strategy at input, but constructing triplets is expensive
- Validation stats (feature size 64):
  - NMI: **0.54**
  - F1 score: **0.39**
- Test stats (feature size 64):
  - NMI: **0.35** (state-of-art pair 0.46, triplet 0.5, lifted 0.54)
  - F1 score: **0.18** (state-of-art pair 0.12, triplet 0.16, lifed 0.19-0.2)



Conclusions & Future Work

Conclusions:

- F1 score close to state-of-the-art, NMI still below
- Loss much more stable when using all hard triplets within a mini-batch

More exploration for architecture with violating triplets sampled at output: Different loss functions, different networks, hyper-parameters

Local Positive Sampling: Sample from a small neighborhood around anchor, to reduce instability

Multiple Points, Learn Manifolds: Learn multiple key points instead of K-means clustering per class

Classification Loss: Improve stability by adding a classification loss using soft-voting from key points

Stanford University