Learning Fine-grained Image Similarity with Deep Ranking

Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, Ying Wu

Let’s start with defining similarity between images…

Similarity between images - Euclidean distance

$$D(p,q) = D(q,p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \ldots + (q_n - p_n)^2}$$

Squared Euclidean distance:

$$D(f(P), f(Q)) = \|f(P) - f(Q)\|_2^2$$

What do we care about image similarity?

Image Search

Query

Ranked Results

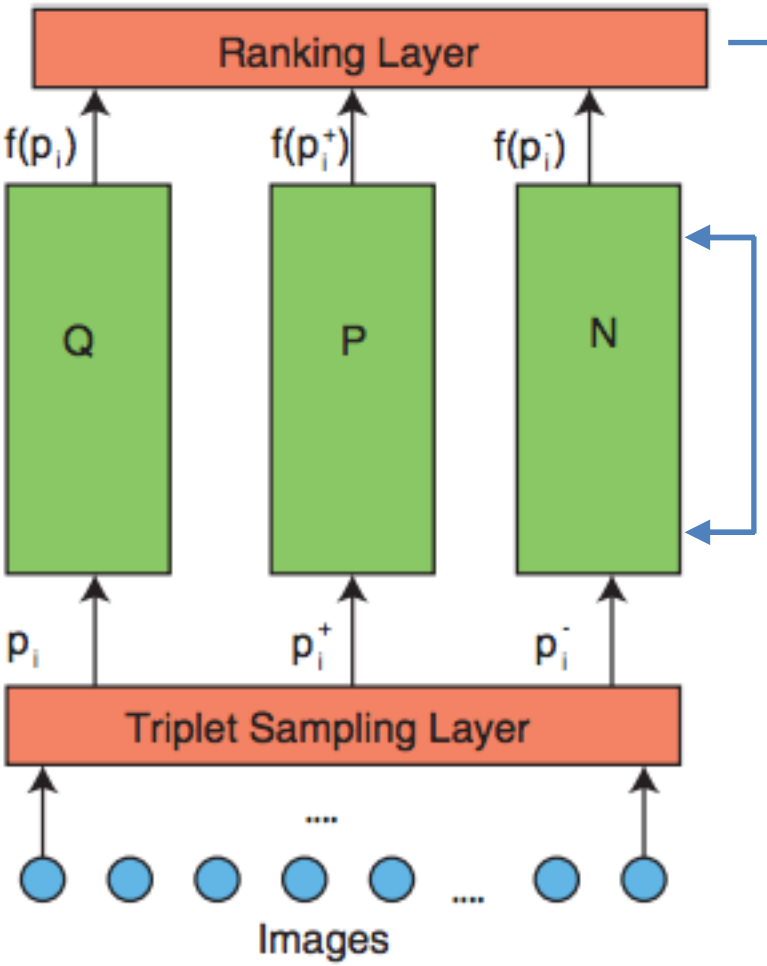
Problem

- Learning image similarity is a challenging problem

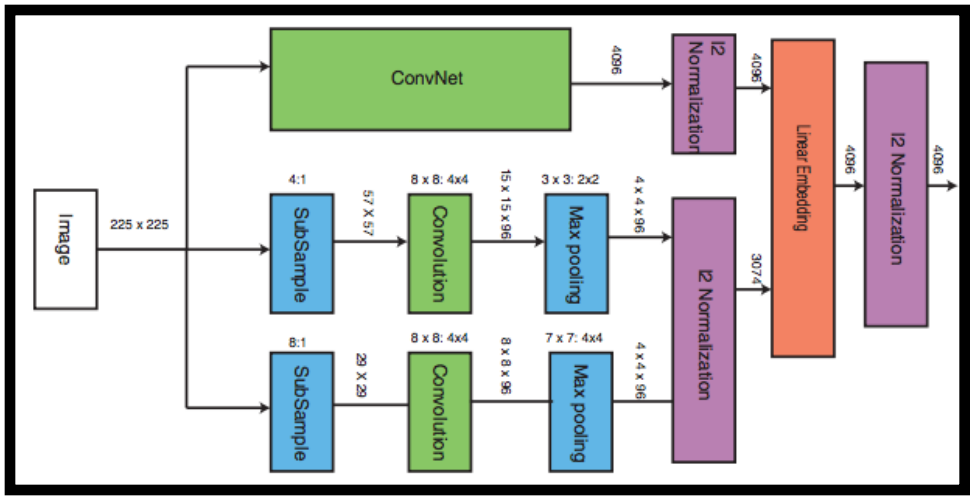
- Most similarity models consider category-level similarity
  - For example, if a query image is a “black car”, we usually want to rank the “dark gray car” higher than the “white car”

We only care about category-level similarity!

Deep Ranking Architecture

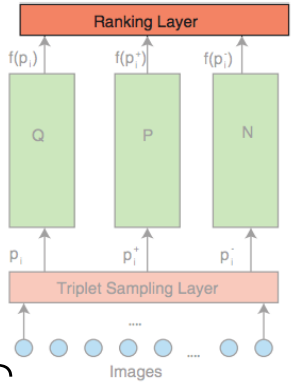


Evaluates Hinge Loss



ResNet

Deep Ranking Goal



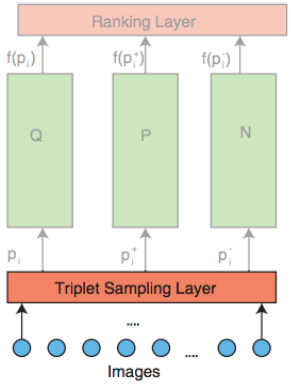
- Learn an embedding function **f(.)** that assigns smaller distance to more similar image pairs

$$D(f(p_i), f(p_i^+)) < D(f(p_i), f(p_i^-)),$$
$$\forall p_i, p_i^+, p_i^- \text{ such that } r(p_i, p_i^+) > r(p_i, p_i^-)$$

- Hinge loss for a triplet

$$l(p_i, p_i^+, p_i^-) =$$
$$\max\{0, g + D(f(p_i), f(p_i^+)) - D(f(p_i), f(p_i^-))\}$$

Triplet Sampling

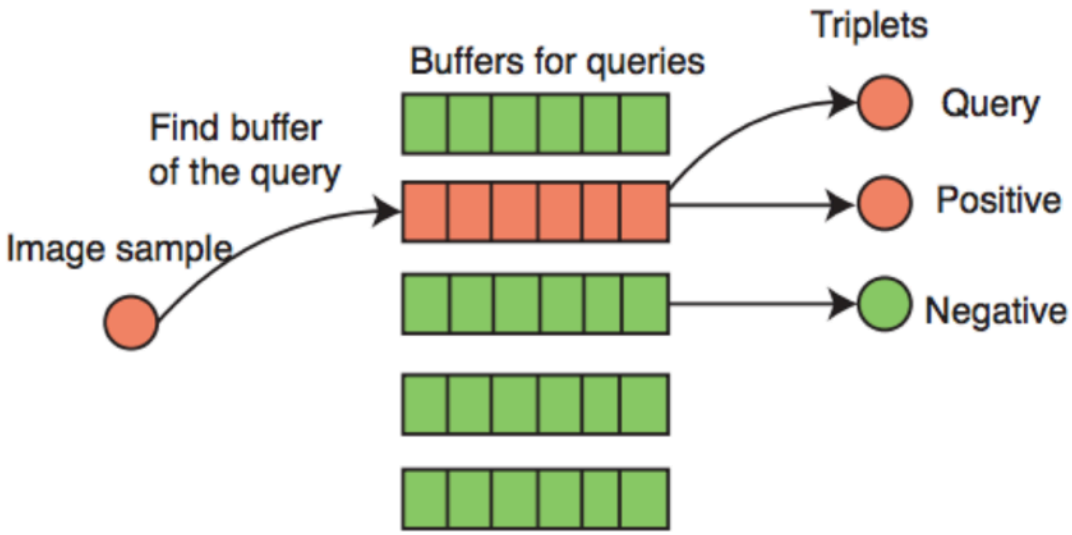


- Need a large variety of images
  - Computationally prohibitive to use all the triplets

- Triplet sampling strategy crucial
  - Uniformly sampling sub-optimal
  - More interested in the top-ranked results returned by the ranking model

You can do uniform sampling for this project!

Triplet Sampling - Uniform sampling



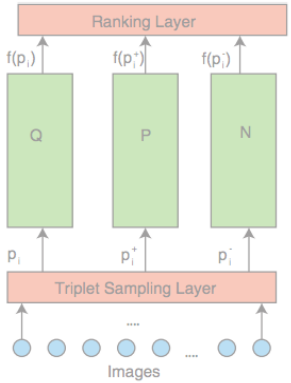
- **Query sample:** p_i is uniformly sampled from all images in the buffer of category c_j
- **Positive image sample:** uniformly sample p^+_i from the same buffer as the query image
- **Out-of-class negative image sample:** draw a image p^-_i uniformly from all the images in the other buffers
- **In-class negative image sample:** not applicable for this homework

Training Data

- **First Dataset**:
  - ImageNet for ConvNet pretraining

- **Second Dataset**:
  - Tiny ImageNet for training and validation

Inference

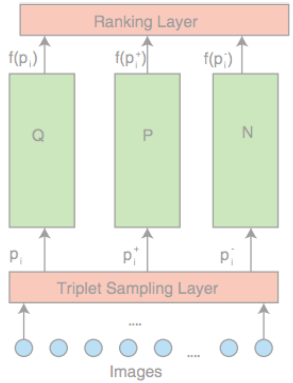


Training Images

Trained CNN

Feature Embeddings

Inference



Query Image

Trained CNN

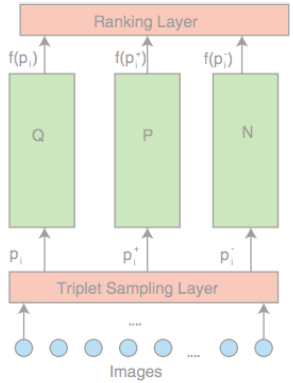
Query Feature Embedding

Training Images

Trained CNN

Feature Embeddings

Inference



Query Image

Trained CNN

Query Feature Embedding

Calculate Euclidean Distances

- Distance between query embedding and all training embeddings

Training Images

Trained CNN

Feature Embeddings

Quantitative Results - Evaluation Criteria

- Retrieve top 30 closest results for a query image
  - Closest in terms of Euclidean distance

- **Accuracy**: How many retrieved images belong to the same class as the query image?

- **Precision** at Top 30:
  - Precision = TPs / (TPs + FPs)
  - Same as accuracy defined above

Qualitative Results

Deliverables

- Code and Accuracy - Target accuracy: 60% or higher
- Describe your implementation
- Quantitative results
  - Plot of your training loss
  - Table of similarity precision for both your training and val
- Qualitative results
  - Sample 5 different images (from different classes) from the val set
    - Show the top 10 ranked results from your pipeline
    - Show the bottom 10 ranked results from your pipeline
- Describe at least one way in how you can improve the performance of your network