Deep Learning for Computer Vision

Image Descriptor Matching

Vineeth N Balasubramanian

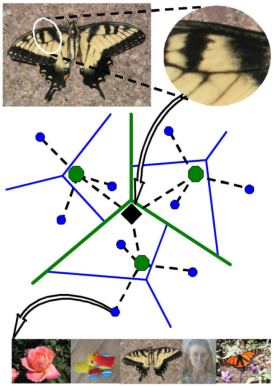
Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad

Acknowledgements

- Most of this lecture's slides are based on lectures of **Deep Learning for Vision** course taught by Prof Yannis Avrithis at Inria Rennes-Bretagne Atlantique

Review[1]

Hierarchical k-means and BoW:



- Apply hierarchical k-means and build a fine partition tree
- Descriptors descend from root to leaves by finding nearest node at each level
- Image represented by $x_i = w_i n_i$ as in BoW
- Dataset searched by inverted files at leaves
- No principled way of defining w_i across levels
- Distortion minimized only locally; points can get assigned to leaves that are not globally nearest

[1] Nister and Stewenius, Scalable Recognition With a Vocabulary Tree, CVPR 2006

Image Descriptor Matching: Options So Far

Nearest Neighbor Matching:

- Use each feature in a set to independently index into second set. Any problems you see?

Image Descriptor Matching: Options So Far

Nearest Neighbor Matching:

- Use each feature in a set to independently index into second set. Any problems you see?
- Ignores possibly useful information of co-occurrence $\Longrightarrow$ fails to distinguish between instances where an object has varying numbers of similar features since multiple features may be matched to a single feature in the other set

Source: Alberto Del Bimbo, UNIFI, Italy

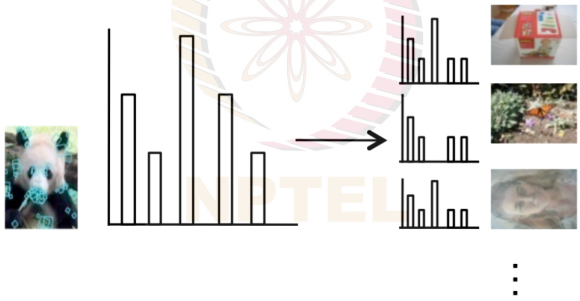
Image Descriptor Matching: Options So Far

Bag-of-Words Matching: Any glaring limitation?

Image Descriptor Matching: Options So Far

Bag-of-Words Matching: Any glaring limitation?

- Can only compare entire images to one another and does not allow partial matchings
- This implies an **all-all** matching; it is often preferable to have a **one-one** matching instead



Source: Alberto Del Bimbo, UNIFI, Italy

Generalizing Descriptor Matching using Kernels[2]

- Consider an image described by a set of n descriptors (features) $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, each of d dimensions

[2] Tolias et al, To Aggregate or Not to aggregate: Selective Match Kernels for Image Search, CVPR 2013

Generalizing Descriptor Matching using Kernels[2]

- Consider an image described by a set of n descriptors (features) $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, each of d dimensions
  - Descriptors typically quantized using k-means clustering
  - Quantizer $q : R^d \rightarrow C \subset R^d$ maps each descriptor to a representative descriptor, a.k.a **visual word**
  - $C = \{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_k\}$ is a codebook consisting of k visual words.

[2] Tolias et al, To Aggregate or Not to aggregate: Selective Match Kernels for Image Search, CVPR 2013

Generalizing Descriptor Matching using Kernels[2]

- Consider an image described by a set of n descriptors (features) $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, each of d dimensions
  - Descriptors typically quantized using k-means clustering
  - Quantizer $q : R^d \rightarrow C \subset R^d$ maps each descriptor to a representative descriptor, a.k.a **visual word**
  - $C = \{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_k\}$ is a codebook consisting of k visual words.
- To compare two image representations X and Y, let us define a general family of matching kernels:

$$K(X, Y) = \gamma(X)\gamma(Y) \sum_{\mathbf{c} \in C} M(X_c, Y_c) \tag{1}$$

where $X_c = \{\mathbf{x} \in X : q(\mathbf{x}) = \mathbf{c}\}$ is the set of descriptors assigned to the same visual word, M is a within-cell matching function, and γ is a normalization function

[2] Tolias et al, To Aggregate or Not to aggregate: Selective Match Kernels for Image Search, CVPR 2013

Bag of Words Matching[3]

Recall: BoW model characterizes an image solely by visual words

- Cosine similarity in BoW model can be defined by defining M as:

$$M(X_c, Y_c) = \sum_{\mathbf{x} \in X_c} \sum_{\mathbf{y} \in Y_c} 1$$

Image Credit: Fei-Fei, Fergus and Torralba, Recognizing and Learning Object Categories, CVPR 2007 Tutorial

[3] Jegou et al, Aggregating local descriptors into a compact image representation, CVPR 2010

Hamming Embedding for Matching[4]

- In addition to being quantized, each descriptor $\mathbf{x}$ is binarized as $\mathbf{b}_x$.
- Score is computed between all pairs of descriptors assigned to the same visual word as:

$$M(X_c, Y_c) = \sum_{\mathbf{x} \in X_c} \sum_{\mathbf{y} \in Y_c} 1[h(\mathbf{b}_x, \mathbf{b}_y) \leq \tau]$$

where $h(\cdot, \cdot)$ is the Hamming distance between two binary vectors, and τ is a threshold to count matched pairs

[4] Jegou et al, Aggregating local descriptors into a compact image representation, CVPR 2010

VLAD Matching[5]

- **Recall:** For each visual word, VLAD performs *pooling* by constructing a vector representing the sum of residuals as:

$$V(X_c) = \sum_{\mathbf{x} \in X_c} r(\mathbf{x}) \quad \text{where} \quad r(\mathbf{x}) = \mathbf{x} - q(\mathbf{x})$$

[5] Jegou et al, Aggregating local descriptors into a compact image representation, CVPR 2010

VLAD Matching[5]

- **Recall:** For each visual word, VLAD performs *pooling* by constructing a vector representing the sum of residuals as:

$$V(X_c) = \sum_{\mathbf{x} \in X_c} r(\mathbf{x}) \quad \text{where} \quad r(\mathbf{x}) = \mathbf{x} - q(\mathbf{x})$$

- A $d \times k$ vector is constructed for an image X as follows:

$$V(X) = (V(X_{c_1}), V(X_{c_2}), V(X_{c_3}), .., V(X_{c_k}))$$

- Matching kernel now defined as:

$$M(X_c, Y_c) = V(X_c)^T V(Y_c) = \sum_{\mathbf{x} \in X_c} \sum_{\mathbf{y} \in Y_c} r(\mathbf{x})^T r(\mathbf{y})$$

[5] Jegou et al, Aggregating local descriptors into a compact image representation, CVPR 2010

Aggregated Selective Match Kernel (ASMK)[6]

- ASMK is a combination of two borrowed ideas:
  - Non-linear selective function (from Hamming Embedding)
  - Pooling Residuals (from VLAD)
- Matching kernel M is expressed as:

$$M(X_c, Y_c) = \sigma_\alpha(\hat{V}(X_c)^T \hat{V}(Y_c))$$

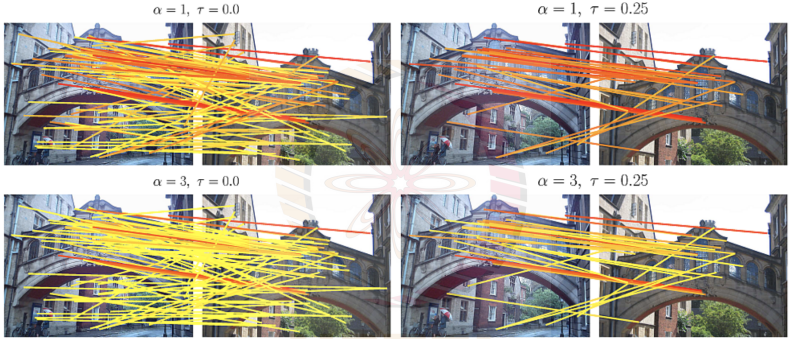
where σ_α is a non-linear function given by:

$$\sigma_\alpha(u) = \begin{cases} \text{sign}(u)|u|^\alpha & \text{if } u > \tau \\ 0 & \text{otherwise} \end{cases}$$

and $\hat{V}(X_c) = V(X_c)/||V(X_c)||$ and $V(X_c)$ is VLAD representation discussed earlier

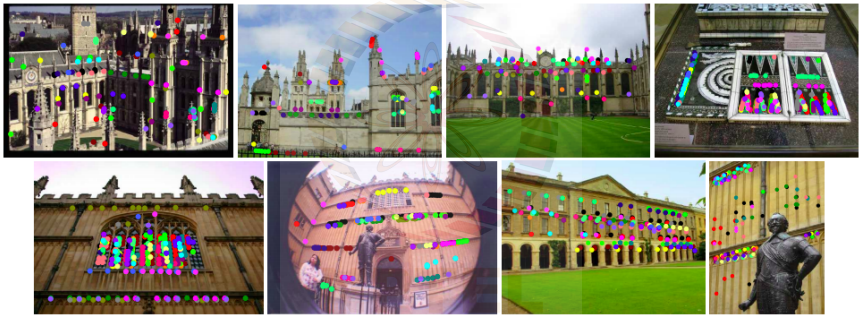
[6] Tolias et al, To Aggregate or Not to aggregate: Selective Match Kernels for Image Search, CVPR 2013

Aggregated Selective Match Kernel (ASMK)



- ASMK matching with different values of distance threshold and selectivity parameter
- Yellow corresponds to 0 similarity and red to maximum similarity per image pair, as defined by selective function
- Larger selectivity drastically down-weights false correspondences
- This replaces hard thresholding in the Hamming Embedding method

Aggregated Selective Match Kernel (ASMK)



ASMK Example: Each visual word is drawn with a different color

Efficient Match Kernels[7]

- Instead of threshold-based matching functions (as used in HE), we can use a continuous function $\kappa(\mathbf{x}, \mathbf{y})$ and avoid using computationally intensive codebooks:

$$K(X, Y) = \gamma(X)\gamma(Y) \sum_{\mathbf{x} \in X} \sum_{\mathbf{y} \in Y} \kappa(\mathbf{x}, \mathbf{y})$$

- Such a function $K(X, Y)$ can be decomposed into an inner product of $\Phi(X)$ and $\Phi(Y)$
- To do that, we learn a low-dimensional feature map ϕ such that $\kappa(x, y) = \phi(x)^T \phi(y)$ and:

$$K(X, Y) = \left(\gamma(X) \sum_{\mathbf{x} \in X} \phi(x) \right)^T \left(\gamma(Y) \sum_{\mathbf{y} \in Y} \phi(y) \right) = \Phi(X)^T \Phi(Y)$$

[7] Bo and Sminchisescu, Efficient Match Kernels between Sets of Features for Visual Recognition, NeurIPS 2009

Homework

Readings

- Tolias *et al.* To Aggregate or Not to aggregate: Selective Match Kernels for Image Search. CVPR 2013
- Chapter 14.4, Szeliski, *Computer Vision: Algorithms and Applications*

References

Liefeng Bo and Cristian Sminchisescu. "Efficient Match Kernels between Sets of Features for Visual Recognition". In: *Proceedings of the 22nd International Conference on Neural Information Processing Systems*. NIPS'09. Vancouver, British Columbia, Canada: Curran Associates Inc., 2009, 135–143.

H. Jégou et al. "Aggregating local descriptors into a compact image representation". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2010, pp. 3304–3311.

Richard Szeliski. *Computer Vision: Algorithms and Applications*. Texts in Computer Science. London: Springer-Verlag, 2011.

G. Tolias, Y. Avrithis, and H. Jégou. "To Aggregate or Not to aggregate: Selective Match Kernels for Image Search". In: *2013 IEEE International Conference on Computer Vision*. 2013, pp. 1401–1408.