Interferences in Match Kernels

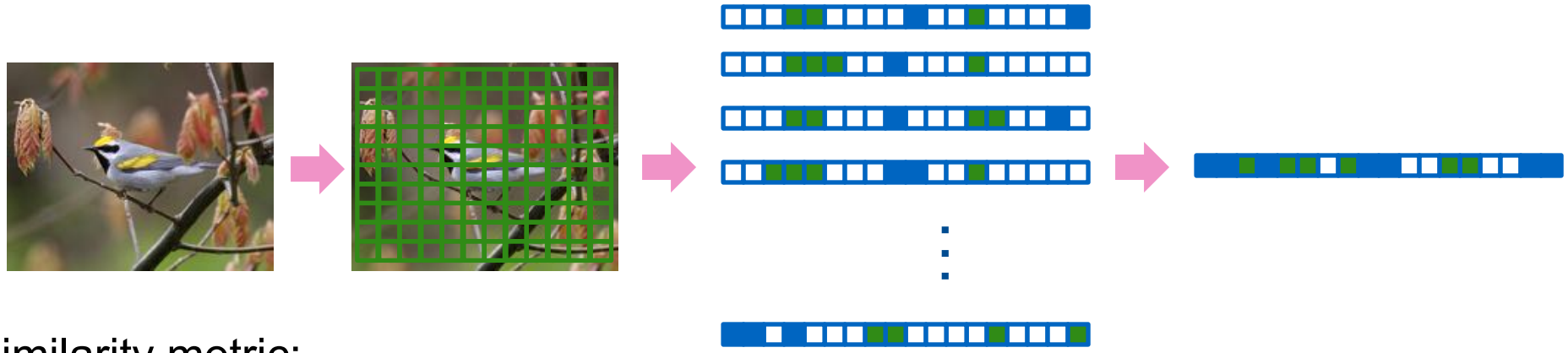
Naila Murray (Naver Labs Europe)
Hervé Jegou (Facebook AI Research)
Florent Perronnin (Naver Labs Europe)
Andrew Zisserman (DeepMind, U. of Oxford)

Horizon Maths, November 23rd, 2018

Patch-based image representations

How to represent images in a patch-based framework? Traditionally:

- extract N patch descriptors $\{x_1, \ldots, x_N\}$
- encode patch descriptor: $x \rightarrow \varphi(x)$
- aggregate encodings



Common similarity metric:
dot-product → can be interpreted as a match kernel
In match kernels, one has to deal with interference, i.e. with the fact that even if two descriptors are unrelated, their matching score may contribute to the overall similarity.

Coding step

Non-linear mapping $x \rightarrow \varphi(x)$ of descriptors into a higher-dim space
e.g. from 128 dim $\rightarrow$ 1K-1M dim

Possible encodings include:

- Bag-of-visual-words (BOV): $\varphi(x) = [0, 0, \ldots, 0, 1, 0, \ldots, 0]'$

  Sivic, Zisserman, “Video Google: A Text Retrieval Approach to Object Matching in Videos”. ICCV 2003.
  Csurka, Dance, Fan, Willamowski, Bray, “Visual categorization with bag of keypoints”. ECCV SLCV 2004.

- Vector of Locally Aggregated Descriptors (VLAD): $\varphi(x) = [0, \ldots, 0, (x - \mu_i), 0, \ldots, 0]'$

  Jégou, Douze, Schmid and Pérez, “Aggregating local descriptors into a compact image representation”. CVPR 2010.

- Fisher Vector (FV): $\varphi(x) = \left[0, 0, \ldots, 0, \frac{(x-\mu_i)}{\sqrt{w_i}\sigma_i}, \frac{1}{\sqrt{2w_i}}\left(\frac{(x-\mu_i)^2}{\sigma_i^2} - 1\right), 0, \ldots, 0\right]'$

  Perronnin and Dance, “Fisher kernels on visual categories for image categorization”. CVPR 2007.

Deep embeddings

Can use direct mapping $I \rightarrow \{\varphi_1, \ldots, \varphi_N\}$ from image to set of patch embeddings, e.g. using CNNs

Possible mapping functions include the convolutional sub-networks of:

- ResNet
  He, Zhang, Ren, Sun. ”Deep Resdiual Learning for Image recognition”. CVPR 2016.
- Dilated Residual Networks
  Yu, Koltun, Funkhouser. “Dilated Residual Networks”. CVPR 2017.
- DELF (DEep Local Features)
  Noh, Araujo, Sim, Weyand, Han. “Largescale image retrieval with attentive deep local features”. ICCV 2017.

→ not the focus of this work, we take embeddings for granted

Aggregation step

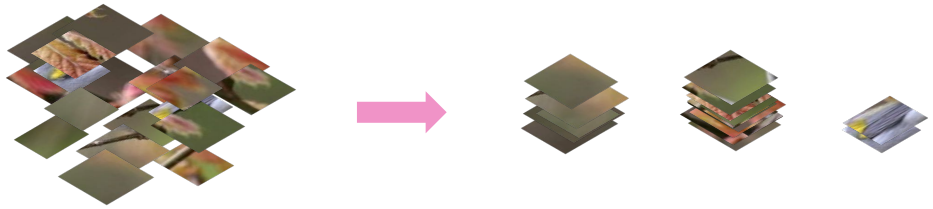
Involves aggregating several patch embeddings:
☺ fixed-length representation
☺ achieves invariance to embedding perturbation
☹ information loss

Aggregation step

Involves aggregating several patch embeddings:
☺ fixed-length representation
☺ achieves invariance to embedding perturbation
☹ information loss

Two standard aggregation strategies:

Sum pooling: $\sum_i \varphi(x_i)$

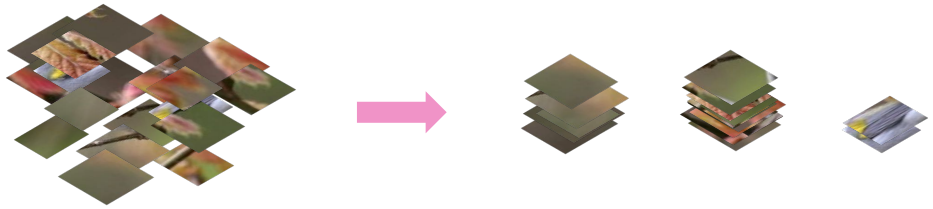


Aggregation step

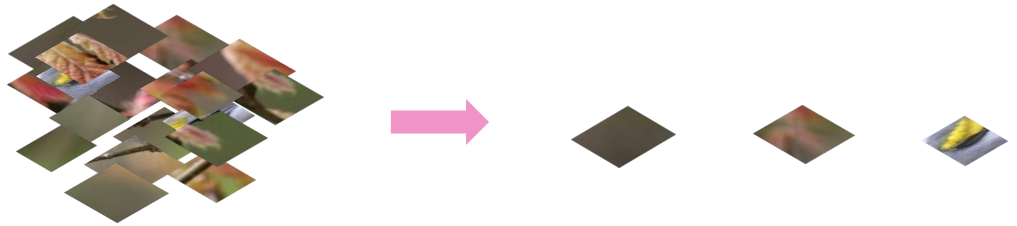
Involves aggregating several patch embeddings:
☺ fixed-length representation
☺ achieves invariance to embedding perturbation
☹ information loss

Two standard aggregation strategies:

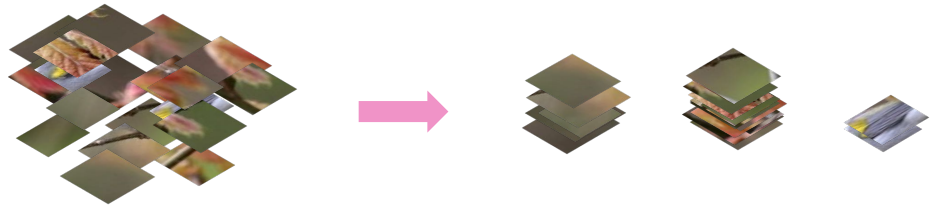
Sum pooling: $\sum_i \varphi(x_i)$



Max pooling: $\max_i \varphi(x_i)$



Sum / Average pooling



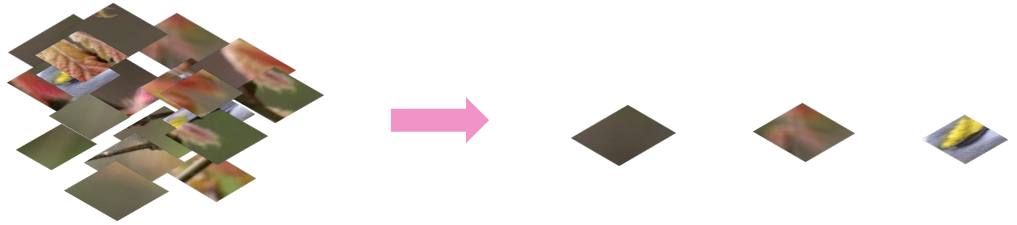
☺ Applicable to any embedding function φ
☹ Based on an incorrect independence assumption
→ frequent (“bursty”) descriptors over-influence the final representation

Corrected a posteriori using non-linear transformations:

- re-weighting visual words
  Jegou, Douze, Schmid, “On the burstiness of visual elements”. CVPR 2009.
- power transformation
  Perronnin, Sanchez, Liu, “Large-scale image categorization with explicit data embedding. CVPR 2010.
  Vedaldi, Zisserman, “Efficient Additive Kernels via Explicit Feature Maps”. CVPR 2010.
  Perronnin, Sanchez, Mensink, “Improving the Fisher kernel for large-scale image classification”. ECCV 2010.
- ℓ_2-normalization over each VLAD pooling region
  Arandjelovic and Zisserman, “All about VLAD”. CVPR 2013.

☹ Heuristic and/or specific to a given representation

Max pooling



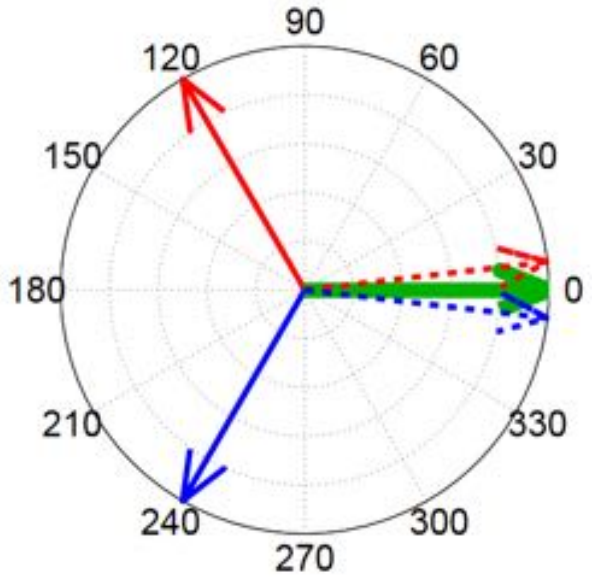
☺ Frequent and rare descriptors contribute meaningfully
☹ Only applicable to BOV-type encodings (e.g. sparse coding)

Application to VLAD, FV, etc. is inappropriate because max operation disregards the encoding structure.

We propose two aggregation strategies that are **applicable to any φ**:
Democratic Aggregation
Generalized Max Pooling

Intuition

We aggregate a single embedding: → or → with a set of tightly clustered embeddings →

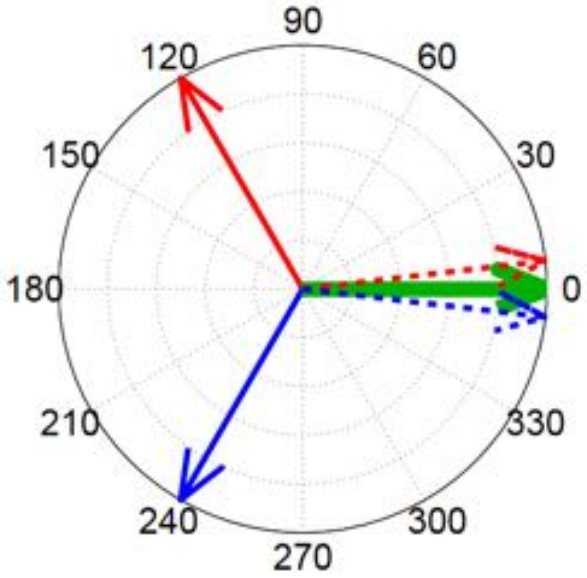


Sum / Average Pooling

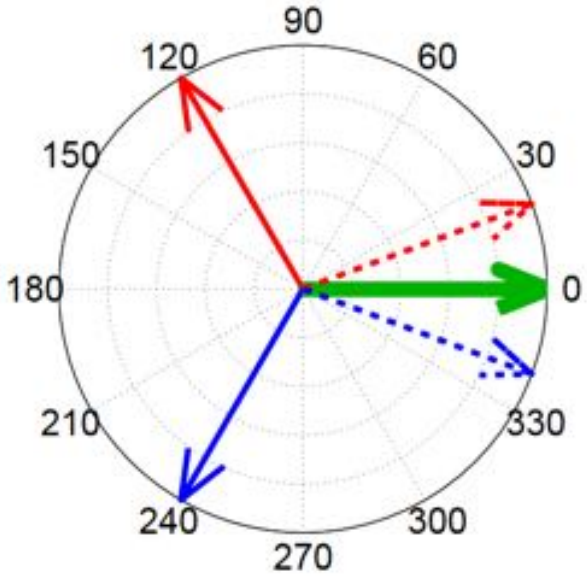
Murray, Jegou, Perronnin, Zisserman. Interferences in match kernels. TPAMI 2017.

Intuition

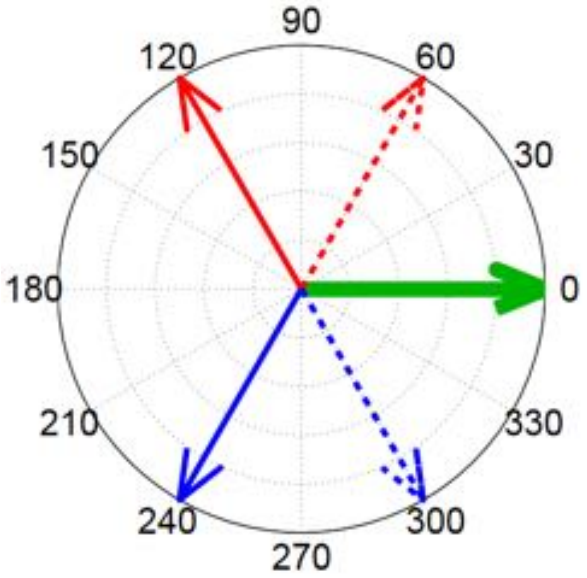
We aggregate a single embedding: → or → with a set of tightly clustered embeddings →



Sum / Average Pooling



Democratic aggregation



Generalized Max Pooling

Murray, Jegou, Perronnin, Zisserman. Interferences in match kernels. TPAMI 2017.

Outline

Introduction

Democratic aggregation

Generalized Max Pooling

Experiments

Conclusion

Outline

Introduction

Democratic aggregation

Generalized Max Pooling

Experiments

Conclusion

Democratic aggregation

Notations:

- φ_i is the $D \times 1$ embedded vector of patch i
- $\Phi = [\varphi_1, \ldots, \varphi_N]$ is the $D \times N$ matrix of embeddings

Democratic aggregation

Notations:

- φ_i is the $D \times 1$ embedded vector of patch i
- $\Phi = [\varphi_1, \ldots, \varphi_N]$ is the $D \times N$ matrix of embeddings

We wish the following to hold for each φ_i:

$$\varphi' \sum_{i=1}^{N} \varphi_i = c$$

→ equal contribution of each patch to self-similarity of the set

Democratic aggregation

Notations:

- φ_i is the $D \times 1$ embedded vector of patch i
- $\Phi = [\varphi_1, \ldots, \varphi_N]$ is the $D \times N$ matrix of embeddings

We wish the following to hold for each φ_i:

$$\varphi' \sum_{i=1}^{N} \varphi_i = c$$

→ equal contribution of each patch to self-similarity of the set

As we typically ℓ_2-normalize the final embedding φ^*, we can set c arbitrarily.

In matrix form, with $K = \Phi' \Phi$ we rewrite: $K\mathbf{1}_N = \mathbf{1}_N$

Democratic aggregation

To achieve democratization we introduce a set of weights $\{\alpha_1, \dots, \alpha_N\}$:

$$\alpha\varphi' \sum_{i=1}^{N} \alpha_i \varphi_i = c$$

In matrix form, with $A = diag(\boldsymbol{\alpha})$, we write: $AKA\mathbf{1}_N = \mathbf{1}_N$

We solve for A using a modified version of the Sinkhorn algorithm.

Outline

Introduction

Democratic aggregation

Generalized Max Pooling

Experiments

Conclusion

Outline

Introduction

Democratic aggregation

Generalized Max Pooling

Experiments

Conclusion

Generalized Max Pooling

Notations:

- φ_i is the $D \times 1$ embedded vector of patch i
- $\Phi = [\varphi_1, \ldots, \varphi_N]$ is the $D \times N$ matrix of embeddings

Generalized Max Pooling

Notations:

- φ_i is the $D \times 1$ embedded vector of patch i
- $\Phi = [\varphi_1, \ldots, \varphi_N]$ is the $D \times N$ matrix of embeddings

We wish our aggregated representation φ^* to satisfy:

$$\varphi_i' \varphi^* = c$$

$\rightarrow$ equalize the matching contribution of each patch

Generalized Max Pooling

Notations:

- φ_i is the $D \times 1$ embedded vector of patch i
- $\Phi = [\varphi_1, \ldots, \varphi_N]$ is the $D \times N$ matrix of embeddings

We wish our aggregated representation φ^* to satisfy:

$$\varphi_i' \varphi^* = c$$

$\rightarrow$ equalize the matching contribution of each patch

Since we typically ℓ_2-normalize the final φ^*, we can set c arbitrarily

In matrix form, we rewrite: $\Phi' \varphi^* = \mathbf{1}$

Generalized Max Pooling

$\Phi'\varphi^* = \mathbf{1}$ is a system of N linear equations with D unknowns
→ may not have a solution (e.g. $D < N$)
→ may have an infinite number of solutions (e.g. $N < D$)

Turn into a least squares regression problem:

$$\varphi^* = \arg\min_{\varphi} \|\Phi'\varphi - \mathbf{1}\|^2$$

under the constraint φ^* has minimal norm

Generalized Max Pooling

$\Phi'\varphi^* = \mathbf{1}$ is a system of N linear equations with D unknowns
→ may not have a solution (e.g. $D < N$)
→ may have an infinite number of solutions (e.g. $N < D$)

Turn into a least squares regression problem:

$$\varphi^* = \arg\min_{\varphi} \|\Phi'\varphi - \mathbf{1}\|^2$$

under the constraint φ^* has minimal norm

→ unique solution: $\varphi^* = (\Phi')^+\mathbf{1} = (\Phi\Phi')^+\Phi\mathbf{1}$

Note that $\Phi\mathbf{1} = \sum_{i=1}^{N} \varphi_i$ is the sum-pooled representation

Relationship with max-pooling

BOV encoding:

- $\varphi_i = [0, 0, \dots, 0, 1, 0, \dots, 0]'$
- Denote by n_k the number of patches assigned to codeword k

$$\varphi^* = (\Phi\Phi')^+\Phi\mathbf{1}$$

$$\Phi\Phi' = \begin{pmatrix} n_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & n_K \end{pmatrix} \qquad \Phi\mathbf{1} = [n_1, \dots, n_K]$$

$$(\Phi\Phi')^+ = \begin{pmatrix} i_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & i_K \end{pmatrix} \text{ with } i_k = \frac{1}{n_k}\ if\ n_k \neq 0, 0 \text{ otherwise}$$

k-th dim of φ^* is 1 if $n_k \neq 0$, 0 otherwise → $\boldsymbol{\varphi^*}$ **is the max-pooled BOV**

Relationship with max-pooling

We can show a more general property:

- assume that $\varphi_i \in \{q_1, \dots, q_K\}$
- $Q = [q_1, \dots, q_K]$ the $D \times K$ codebook matrix is orthonormal
- n_k the number of patches such that $\varphi_i = q_k$

We have:

$$\varphi^* = \sum_{k=1, n_k \neq 0}^{K} q_k$$

i.e. **$\boldsymbol{\varphi^*}$ is independent of $\boldsymbol{n_k}$'s**

Regularization

In practice, the pseudo-inverse is not a continuous operation
→ add a regularization parameter λ: $\varphi_\lambda^* = \arg\min_\varphi \|\Phi'\varphi - \mathbf{1}\|^2 + \lambda\|\varphi\|^2$

$$\varphi_\lambda^* = (\Phi\Phi' + \lambda I)^{-1}\Phi\mathbf{1}$$

Not only regularization parameter:

- when $\lambda \to 0$: max pooling
- when $\lambda \to \infty$, $\varphi_\lambda^* \approx \Phi\mathbf{1}/\lambda$: sum pooling

→ interpolation between sum pooling and max pooling

BOV example: $\varphi_\lambda^* = \left[\frac{n_1}{n_1+\lambda}, \ldots, \frac{n_D}{n_D+\lambda}\right]$

Computing the GMP in practice

We need to solve: $(\Phi\Phi' + \lambda I)\varphi_\lambda^* = \Phi\mathbf{1}$
→ linear system of D equations with D unknowns: cost in $O(D^2)$
☹ Impractical if the embedding is high-dimensional

Since $(\Phi\Phi' + \lambda I)$ is PSD, we can use Conjugate Gradient Descent
→ still too slow for large D

Computing the GMP in practice

We need to solve: $(\Phi\Phi'+\lambda I)\varphi_\lambda^* = \Phi\mathbf{1}$
→ linear system of D equations with D unknowns: cost in $O(D^2)$
☹ Impractical if the embedding is high-dimensional

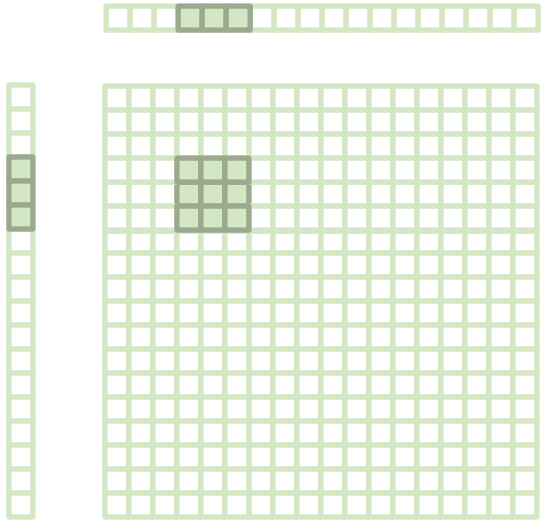
Since $(\Phi\Phi'+\lambda I)$ is PSD, we can use Conjugate Gradient Descent
→ still too slow for large D

Exploit the embedding structure:

- VLAD or FV (with hard assignment) are block-sparse
- $(\Phi\Phi'+\lambda I)$ is block-diagonal

→ solve block-by-block: cost in $O\left(\frac{D}{C}\frac{D}{C}C\right) = O\left(\frac{D^2}{C}\right)$

where C is the codebook size



Outline

Introduction

Democratic aggregation

Generalized Max Pooling

Experiments

Conclusion

Outline

Introduction

Democratic aggregation

Generalized Max Pooling

Experiments

Conclusion

Experiments: instance-level image retrieval

Task: Given a query image, find similar images in a database

Experiments: instance-level image retrieval

Oxford dataset

INRIA Holidays dataset

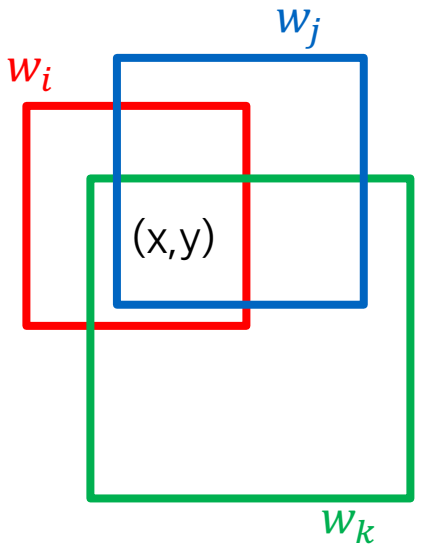
Experiments: instance-level image retrieval

method ↓	$\|C\|$	D	mAP Holidays	mAP Ox5k	mAP Ox105k
BOW [4]	20k	20,000	43.7	35.4	–
BOW [4]	200k	200,000	54.0	36.4	–
VLAD [4]	64	4,096	55.6	37.8	–
Fisher [4]	64	4,096	59.5	41.8	–
VLAD-intra [48]	256	32,536	65.3	55.8	–
VLAD-intra [48]	256	→ 128	62.5	44.8	37.4
Our methods					
$\phi_\triangle + \psi_s$ + RN	16	1,920	68.5	53.7	46.2
$\phi_\triangle + \psi_s$ + RN	64	8,064	73.4	63.0	55.0
$\phi_\triangle + \psi_d$ + RN	16	1,920	70.7	57.4	50.7
$\phi_\triangle + \psi_d$ + RN	64	8,064	75.5	67.0	60.2
$\phi_\triangle + \psi_{gmp}$ + RN	16	1,920	67.1	58.3	51.4
$\phi_\triangle + \psi_{gmp}$ + RN	64	8,064	**76.5**	**70.0**	**64.4**

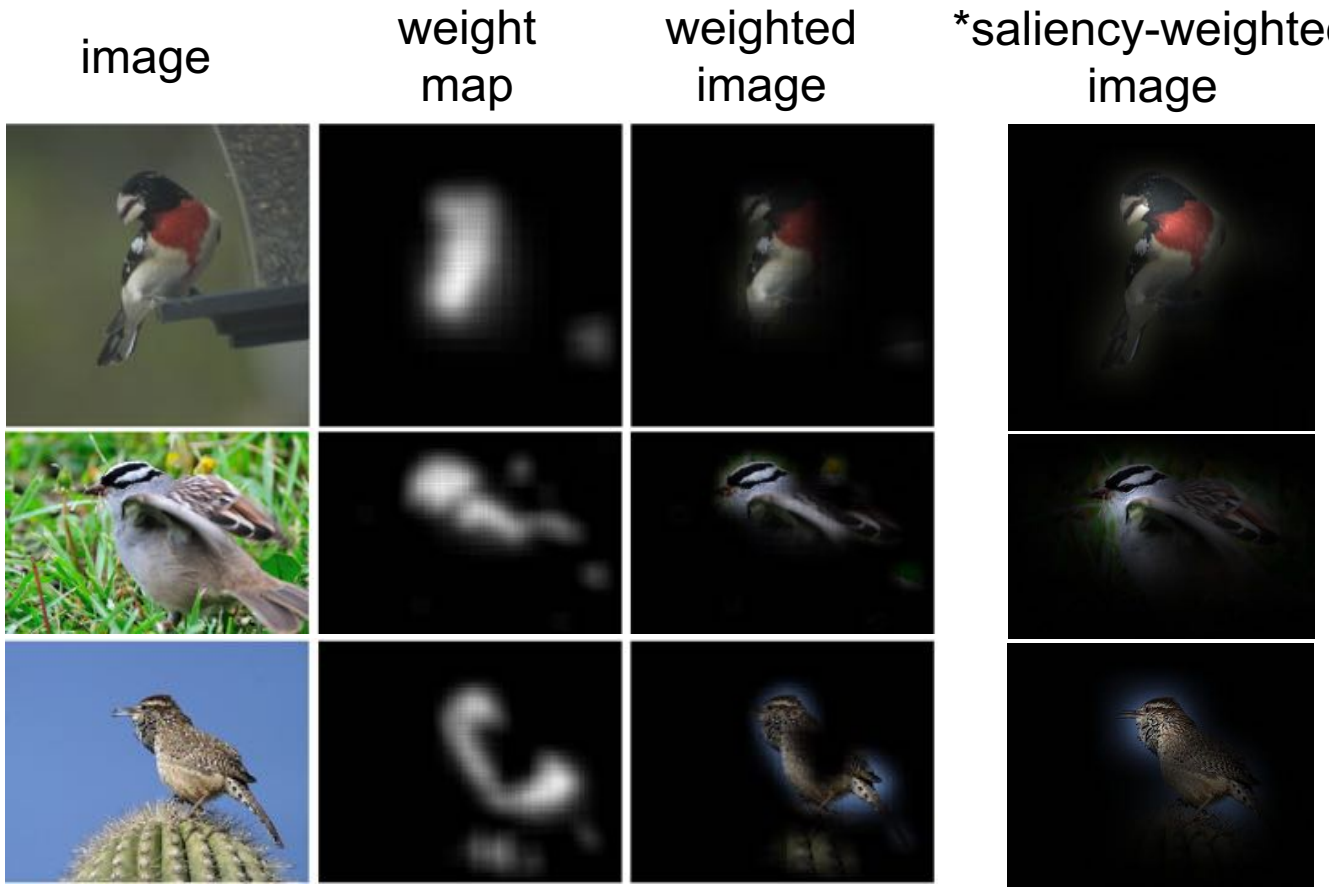
Murray, Jegou, Perronnin, Zisserman. Interferences in Match Kernels. TPAMI, 2016

Relationship with saliency

Compute a weight map: for each pixel, the weight is the sum of the weights of the patches it belongs to



weight at position (x,y) is $w_i + w_j + w_k$



*Jetley, Murray, Vig. End-To-End Saliency Mapping via Probability Distribution Prediction. CVPR, 2016

Outline

Introduction

Democratic aggregation

Generalized Max Pooling

Experiments

Conclusion

Outline

Introduction

Democratic aggregation

Generalized Max Pooling

Experiments

Conclusion

Conclusion

We proposed two aggregation strategies that are applicable to any embedding function φ:

- DA: “democratises” the contribution of each descriptor to a set comparison metric
- GMP: equalises the similarity between each descriptor and the aggregated representation
- Both lead to significant improvements over aggregation baselines

Thanks!