Comprendre les données visuelles à grande échelle

ENSIMAG
2019-2020

Karteek Alahari & Diane Larlus

17 octobre 2019

Organisation du cours

- **17/10/19** cours 1 - Diane
- **24/10/19** cours 2 - Karteek
- **07/11/19** cours 3 - Karteek
- **14/11/19** cours 4 - Diane
- **28/11/19** cours 5 - Karteek
- **05/12/19** cours 6 - Karteek
- **12/12/19** cours 7 - Diane
- **19/12/19** cours 8 - Diane

Vacances d'hiver

- **09/01/20** cours 9 - Diane + présentation articles 1 & 2 + quizz
- **16/01/20** cours 10 - Diane + présentation articles 3 & 4 + quizz
- **23/01/20** cours 11 - Karteek + présentation articles 5 & 6 + quizz
- **30/01/20** cours 12 - Karteek + présentation articles 7 & 8 + quizz

Attention: la salle change régulièrement

NAVER LABS EUROPE

Cours 4: Recherche d’images - suite et fin

Comprendre les données visuelles à grande échelle
14 novembre 2019

Composant clé de la recherche d’images: Indexation

Comprendre les données visuelles à grande échelle

Cours 3: représentations d’images, 10 janvier 2019

NAVER LABS EUROPE

Malédiction de la dimension

ou « fléau de la dimension » (*curse of dimensionality*)

Quelques propriétés surprenantes

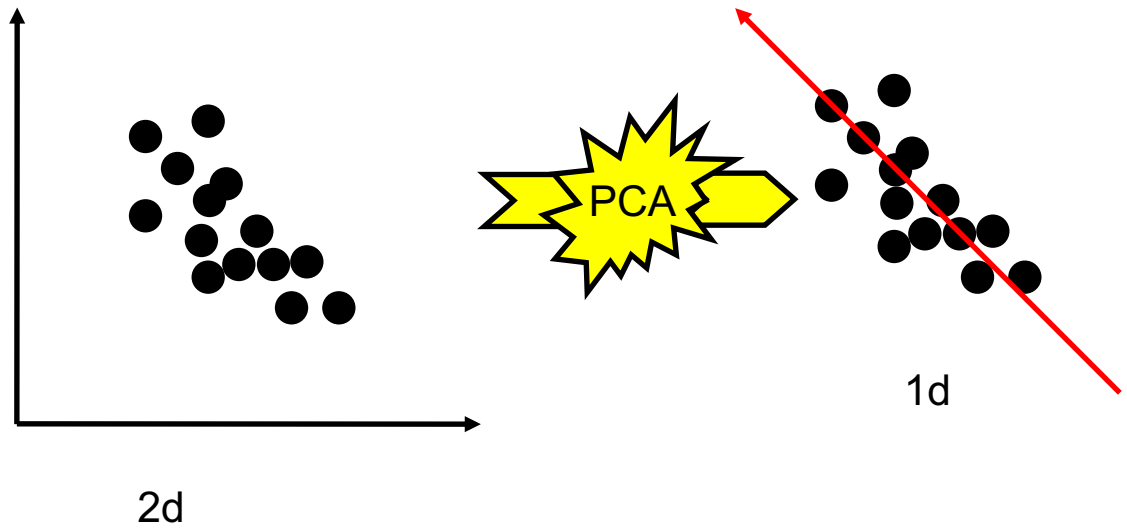
- *vanishing variance:* la distance entre paires de points tend à être identique quand *d* augmente
- phénomène de l'espace vide: si on discrétise l'espace, la plupart des cellules sont vides
- proximité des frontières: la plupart des points d'une hyper-sphère sont « proches » de la frontière

Réduction de la dimensionnalité

- Approche la plus courante :
  analyse en composantes principales (ACP)
  - PCA en anglais : *Principal Component Analysis*

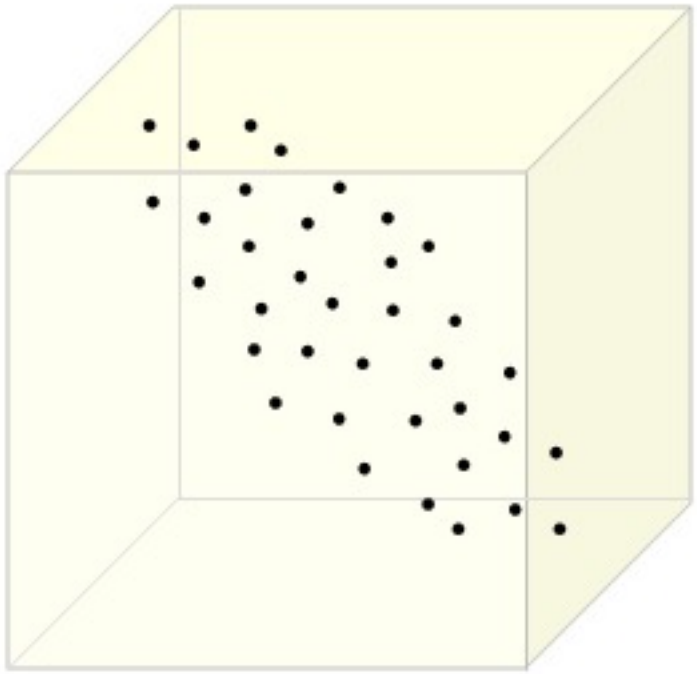
ACP

- Analyse des relations statistiques entre les différentes composantes
- Pour être capable de reproduire la plus grande partie de l'énergie d'un vecteur avec un nombre plus faible de dimension
  - élimination des axes peu énergétiques

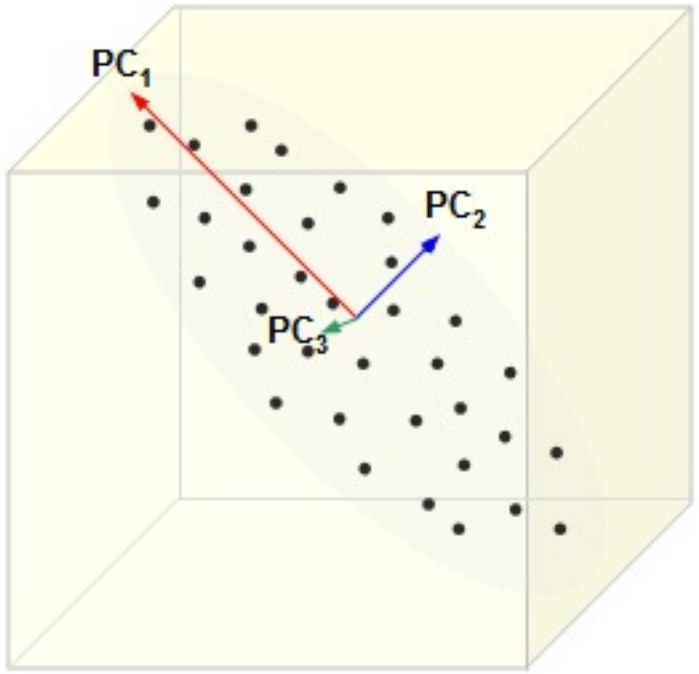


ACP

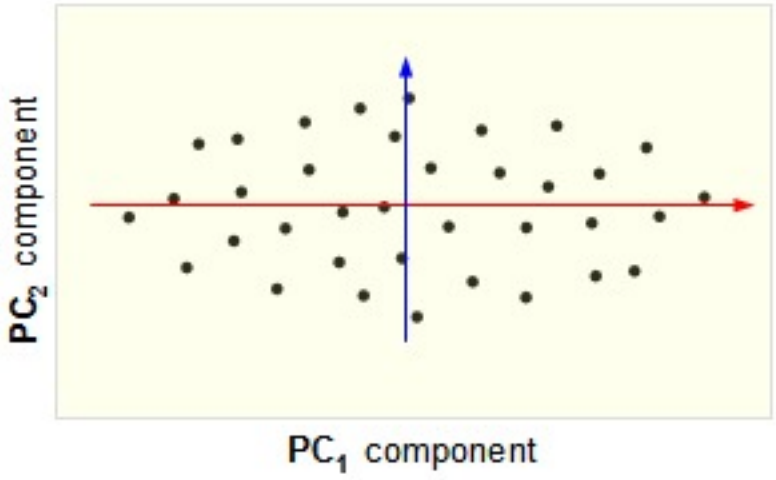
Exemple 3D vers 2D



a



b



c

ACP

- Il s'agit d'un changement de base (translation + rotation)

- On suppose N descripteurs $\{x_1, x_2, x_3, .., x_N\}$
- 4 étapes (*offline*)
  - Calcul de la moyenne des descripteurs
    - $\bar{x} = \frac{1}{N}\sum_1^N x_i$
  - Calcul de la matrice de covariance sur les vecteurs centrés
    - $C = \sum_1^N y_i y_i^T$ avec $y_i = x_i - \bar{x}$
  - Calcul des valeurs propres et vecteurs propres de la matrice de covariance
    - $Ce_i = \lambda_i e_i$
  - Choix des *d'* composantes les plus énergétiques (+ grandes valeurs propres)

- Pour un vecteur, les nouvelles coordonnées sont obtenues par centrage et multiplication du vecteur par la matrice *d' x d* des vecteurs propres conservés

ACP

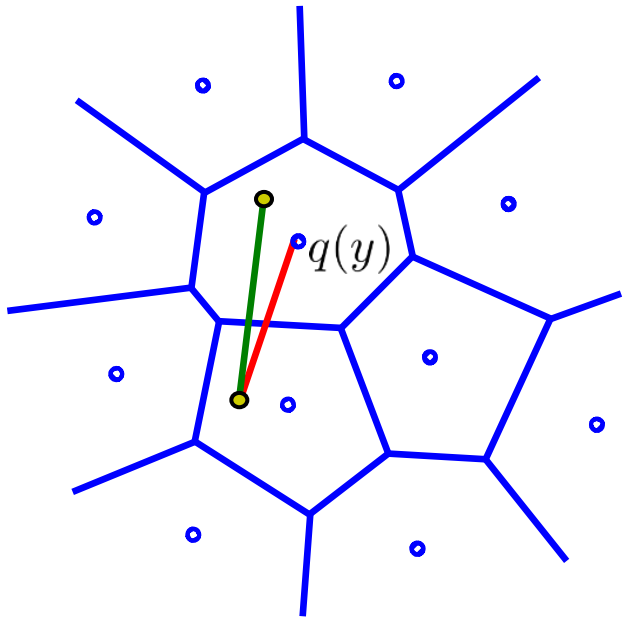
- Moins de dimensions, donc meilleur comportement des algorithmes de recherche
- Si la réduction préserve la plupart de l'énergie
  - les distances sont préservées
  - donc le + proche voisin dans l'espace transformé est aussi le plus proche voisin dans l'espace initial

- Limitations
  - utile seulement si la réduction est suffisamment forte
  - lourdeur de mise à jour de la base (choix fixe des vecteurs propres préférable)
  - peu adapté à certains types de données

Améliorer la finesse de la quantification: le quantificateur produit

$$d(x, y) \approx d(x, q(y))$$

- Problème d'approximation de distances entre requête x et vecteur y de la base
- il faut une quantification plus fine que quelques milliers de centroides
  - *k-means (*l'algorithme des k-moyennes*)* coûteux
  - assignement coûteux
  - stockage des centroïdes coûteux (même en hiérarchique)
- Solution: le quantifieur produit (*product quantization* ou PQ)

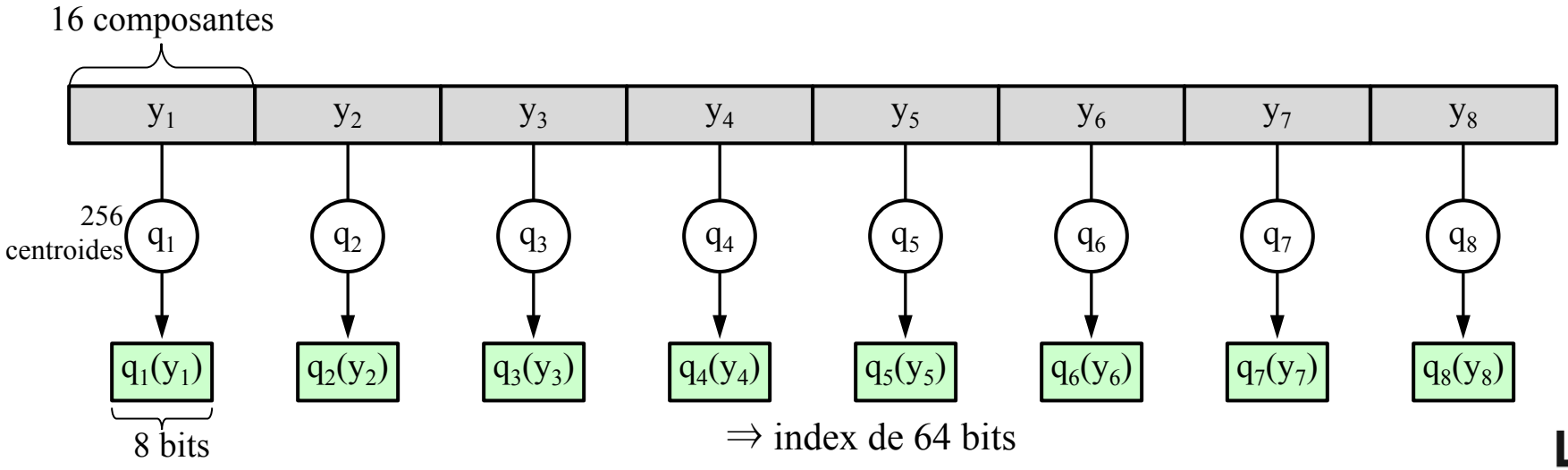


Quantificateur produit pour la recherche des plus proches voisins

- Vecteur coupé en *m* sous-vecteurs: $y \rightarrow [y_1|\ldots|y_m]$
- Sous-vecteurs quantifiés séparément: $q(y) = [q_1(y_1)|\ldots|q_m(y_m)]$

  chaque *qi* a un petit vocabulaire

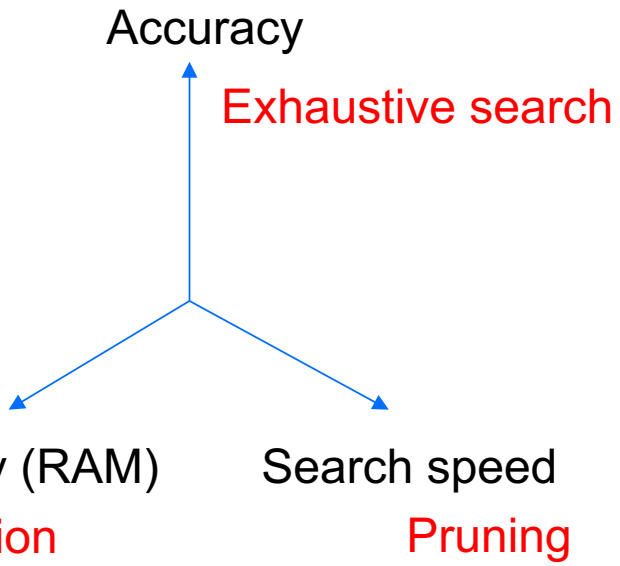
- Exemple de SIFT: y possède 128 dimensions. Il est découpé en 8 sous-vecteurs de dimension 16



NAVER LABS EUROPE

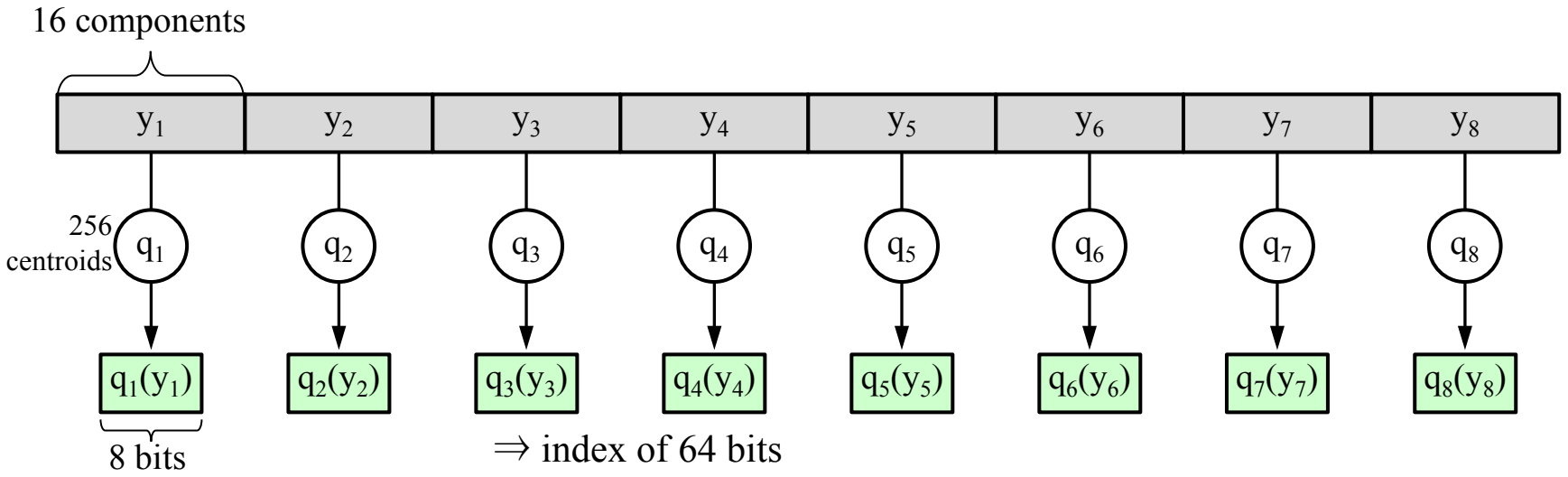
Ingredients of a full system 1/2

Trade-offs in similarity search



Compression

- Product Quantization (PQ)



[Hervé Jegou's tutorials]

NAVER LABS EUROPE

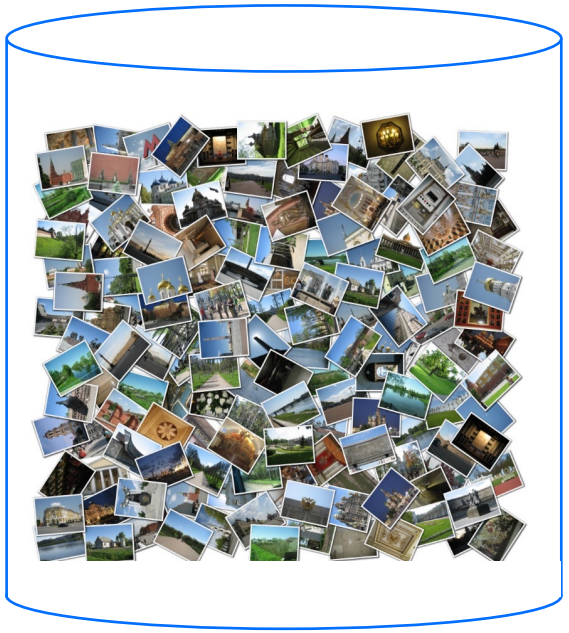
Ingredients of a full system 2/2

Leveraging the topology of the gallery set

- Query expansion

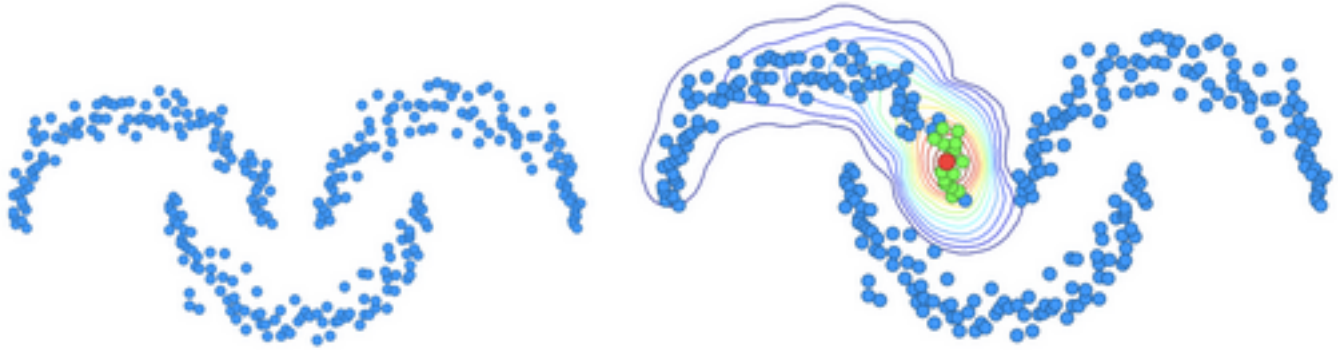
[Chum et al. ICCV07, Chum et al CVPR11]

  - Process the list of results (e.g. spatially verify)
  - If some images are good, use them to process some other augmented queries



- Diffusion

[Iscen et al. CVPR17, CVPR18]



NAVER LABS EUROPE

Requêtes complexes: cross-modalité et recherches sémantiques

Comprendre les données visuelles à grande échelle

Cours 4: recherche d'images, 14 novembre 2019

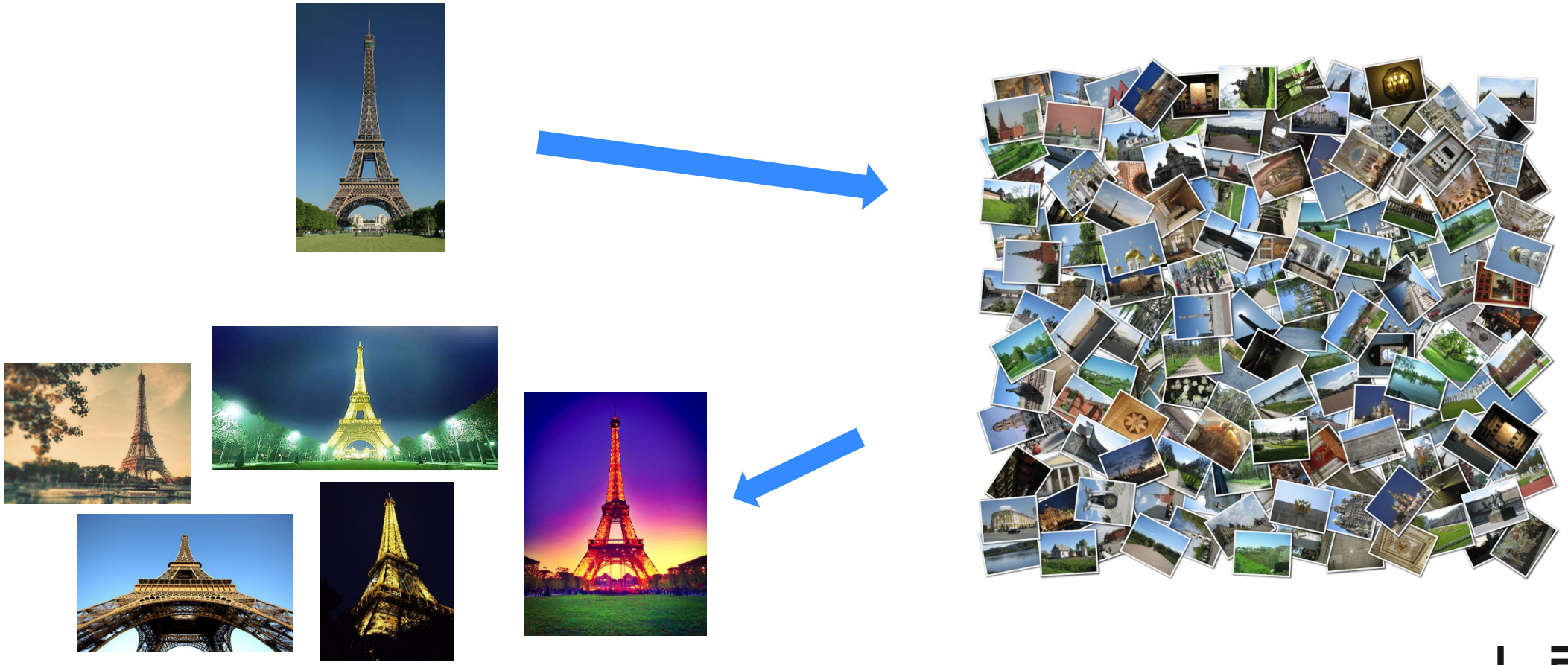
NAVER LABS EUROPE

Plan de la suite

- Recherche visuelle sémantique
  - Principe
  - Apprentissage d’une représentation dédiée
- Recherche cross-modale
  - Notion de plongement (*Embedding*)

La recherche visuelle classique

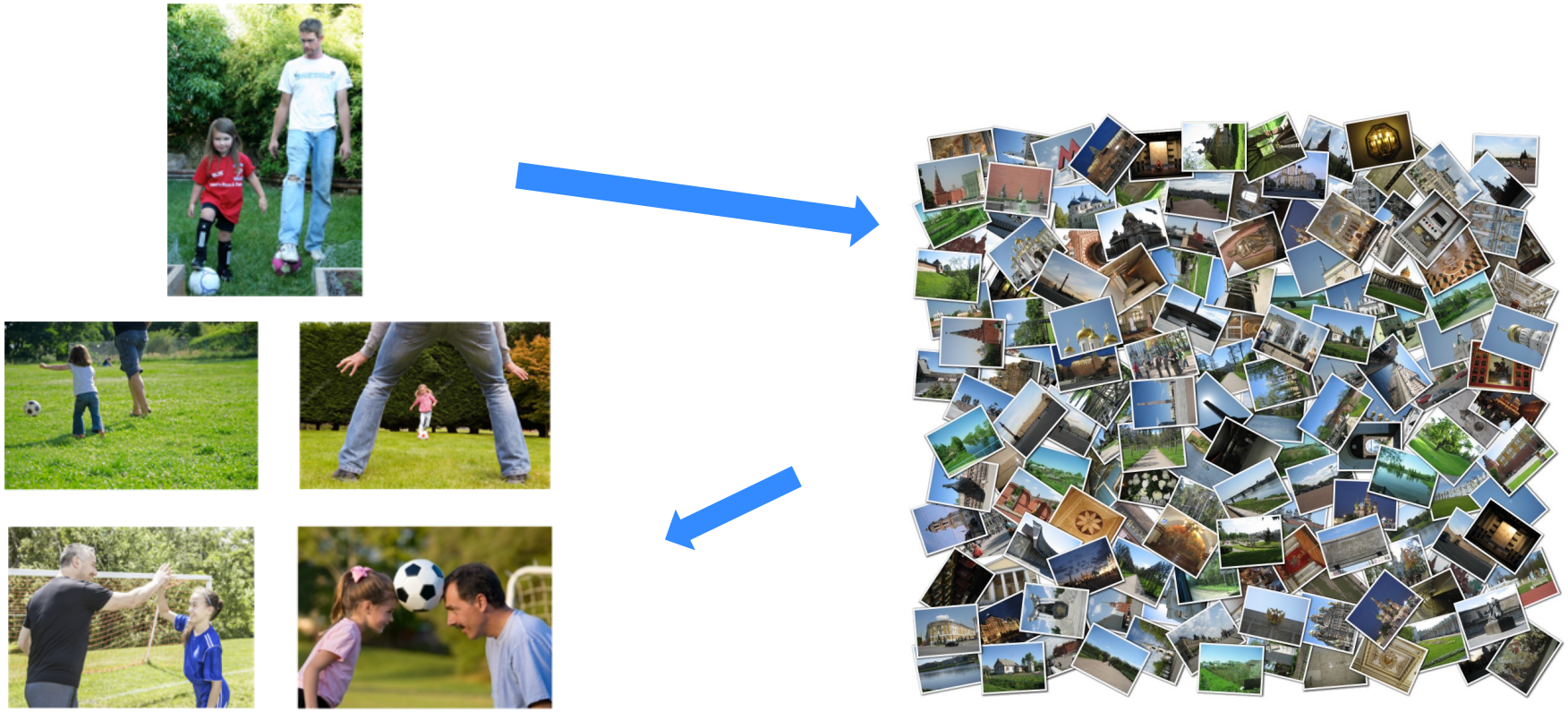
En général, la recherche d'images s'intéresse principalement à la recherche d'instances



Au delà de la recherche d'instances

Supposons que l'on s'intéresse à des **requêtes complexes** qui regardent toute la scène visuelle

- C'est ce qu'on appellera la recherche visuelle sémantique (ou *semantic retrieval*)

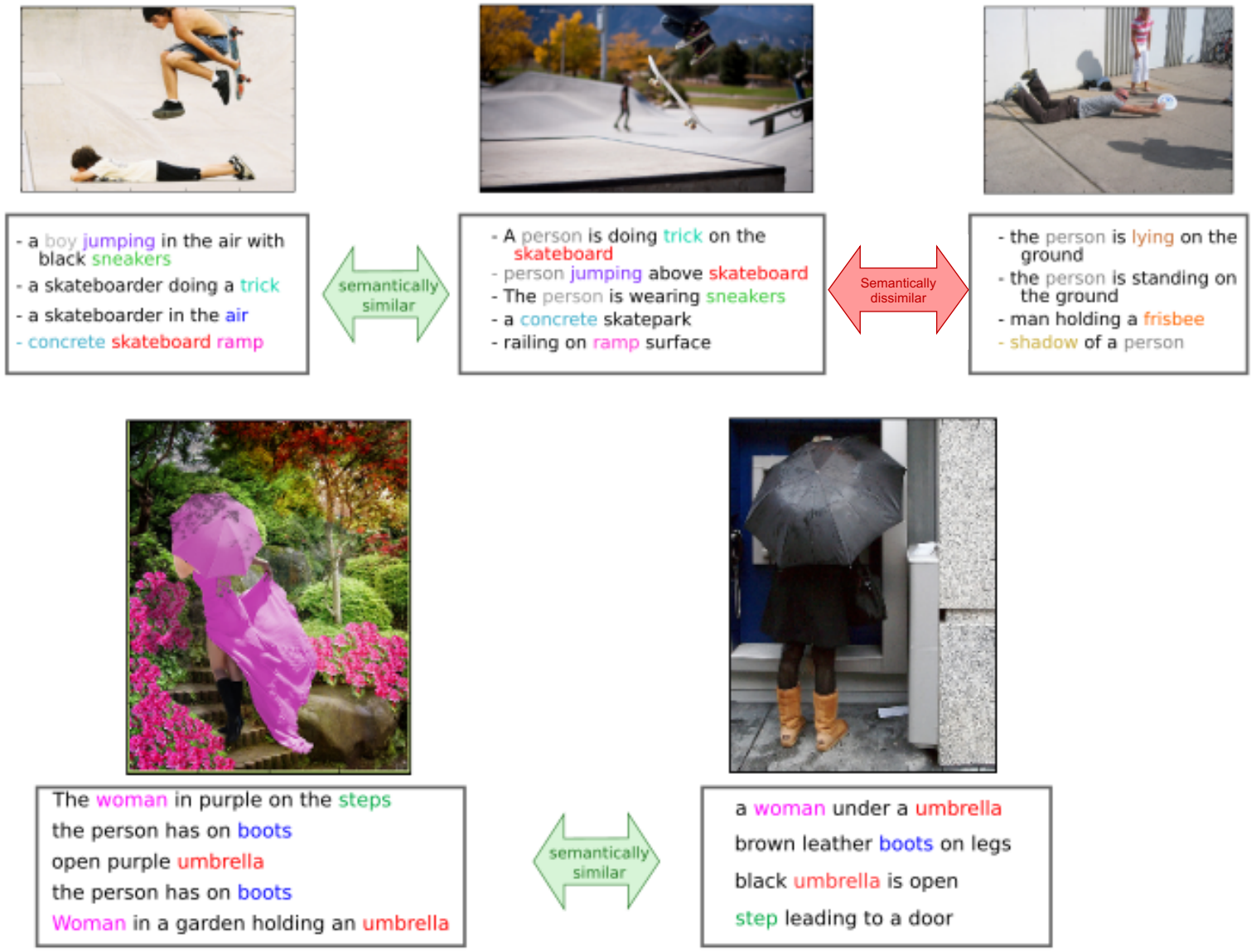


[*Beyond instance-level image retrieval: Leveraging captions to learn a global visual representation for semantic retrieval. Gordo & Larlus. CVPR17*]

NAVER LABS EUROPE

Semantic image retrieval

Human-generated captions



[*Gordo & Larlus. CVPR17*]

NAVER LABS EUROPE

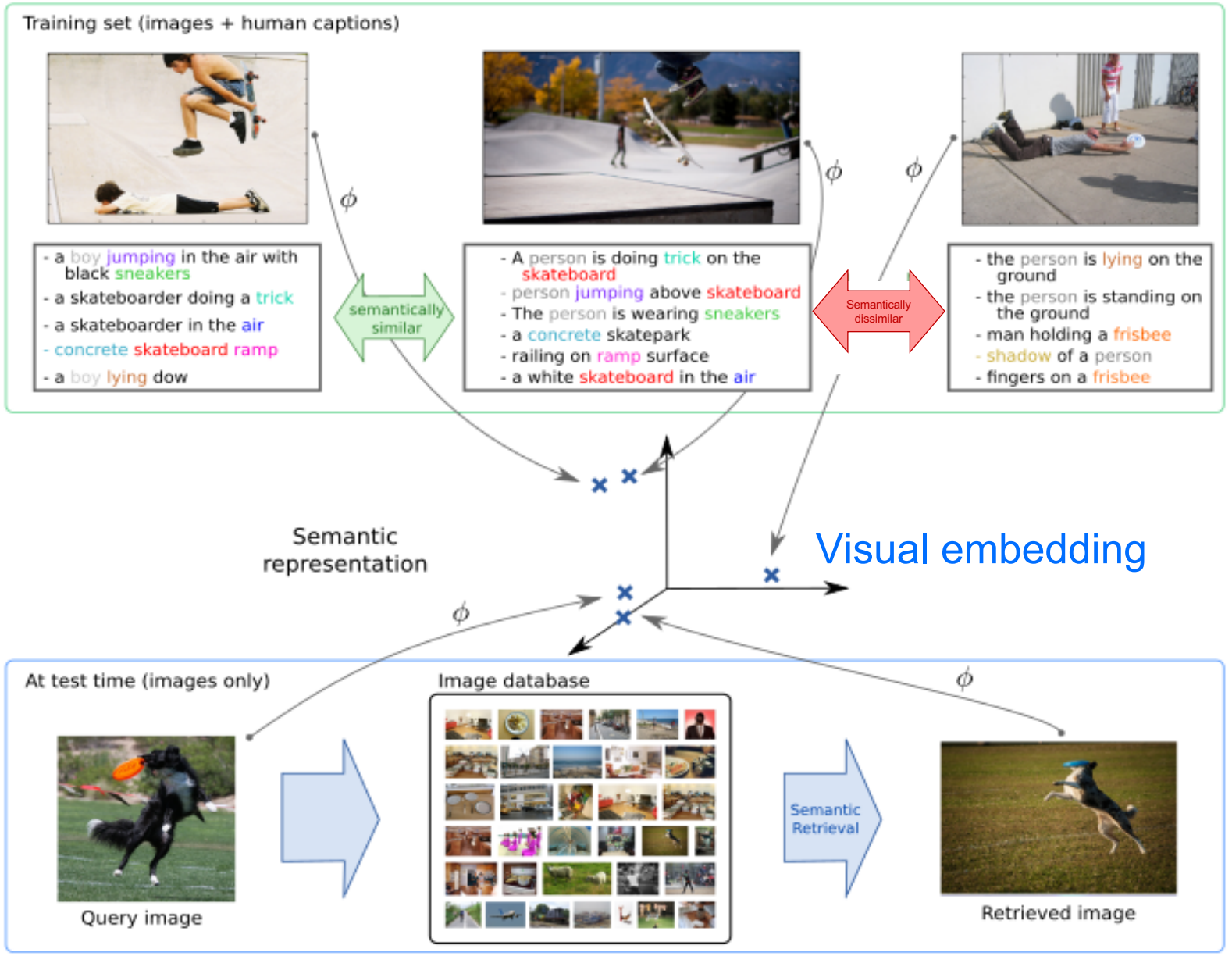
Semantic image retrieval

Building visual representations:

Intuition

For images with similar captions:

Visual representation close in the **semantic embedding space**

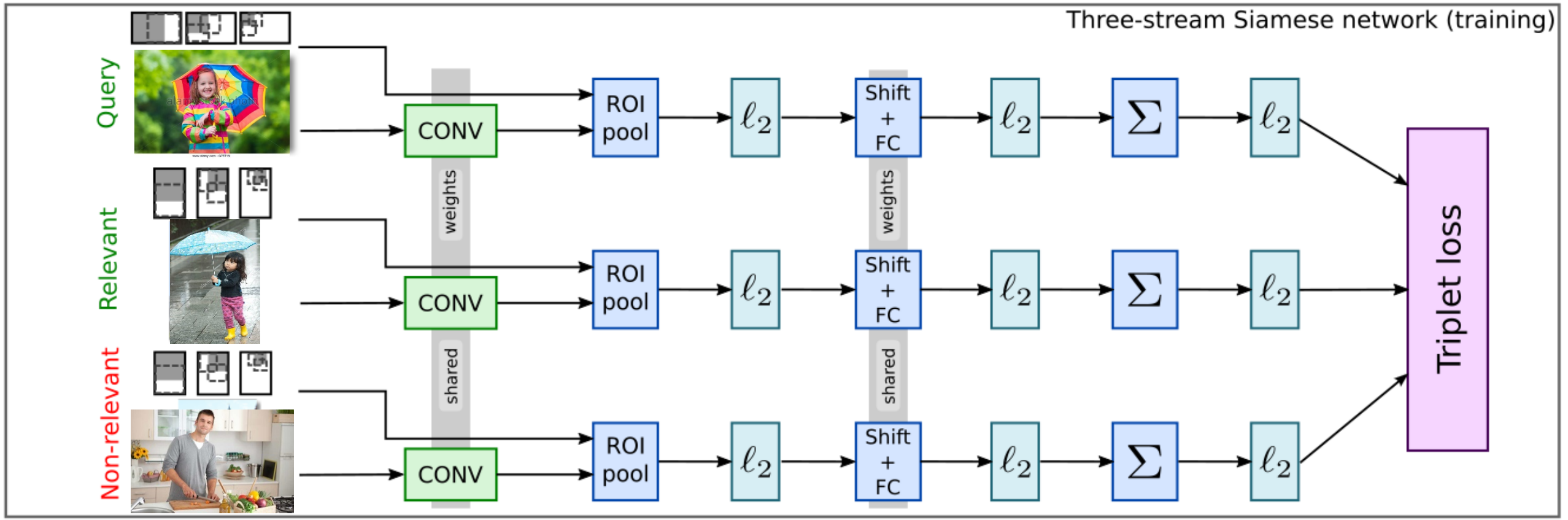


[*Gordo & Larlus. CVPR17*]

Learning a semantic embedding

- Learning to rank
  - Three-stream Siamese Network

[Gordo@ECCV16, Gordo@IJCV17]



Learning a semantic embedding

Visual loss:

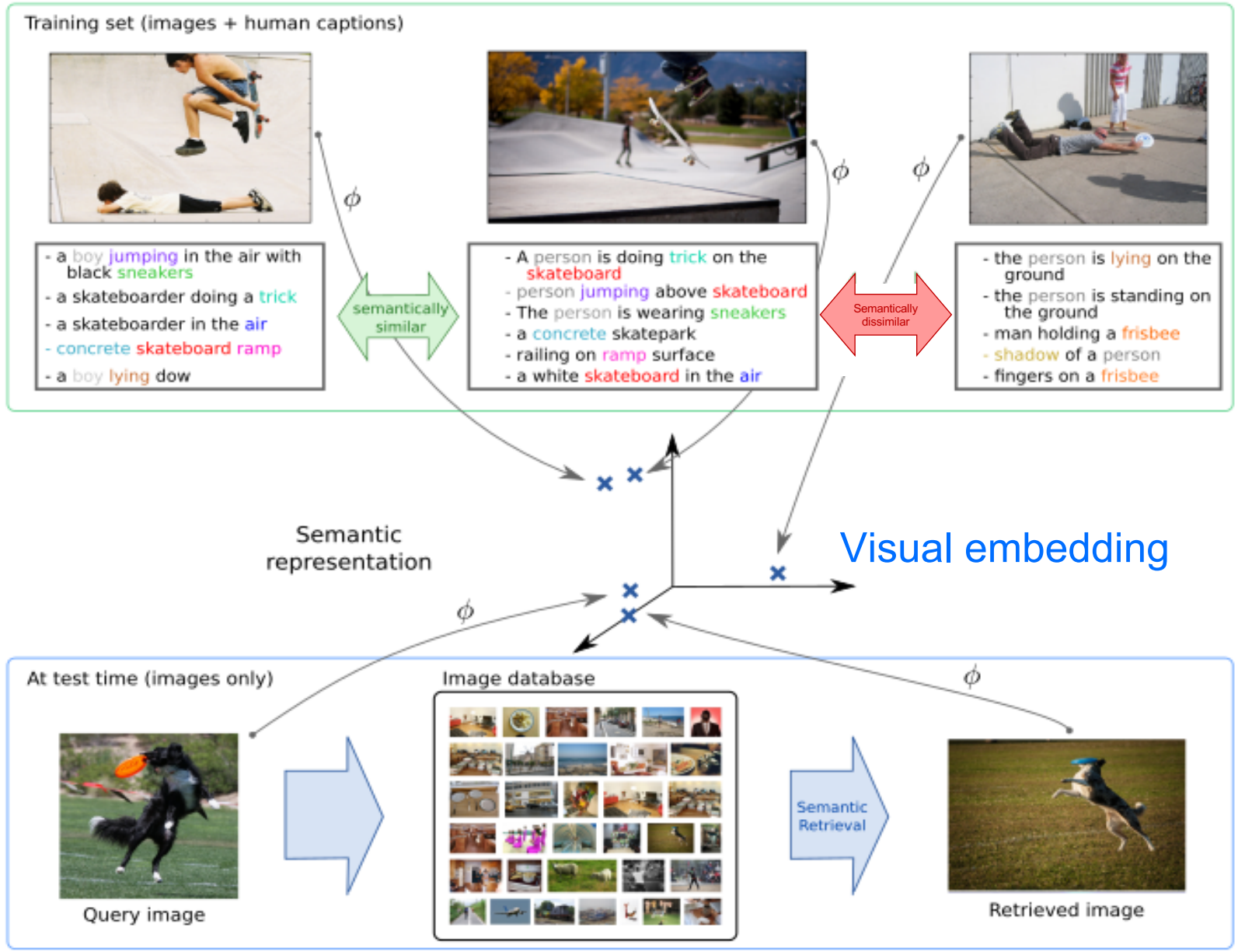
$$L_v(q, d^+, d^-) = \frac{1}{2}\max(0, m - \phi_q^T\phi_+ + \phi_q^T\phi_-)$$

[*Gordo & Larlus. CVPR17*]

Semantic image retrieval

Training the model

The visual representation of images with similar captions are close in the **semantic embedding space**



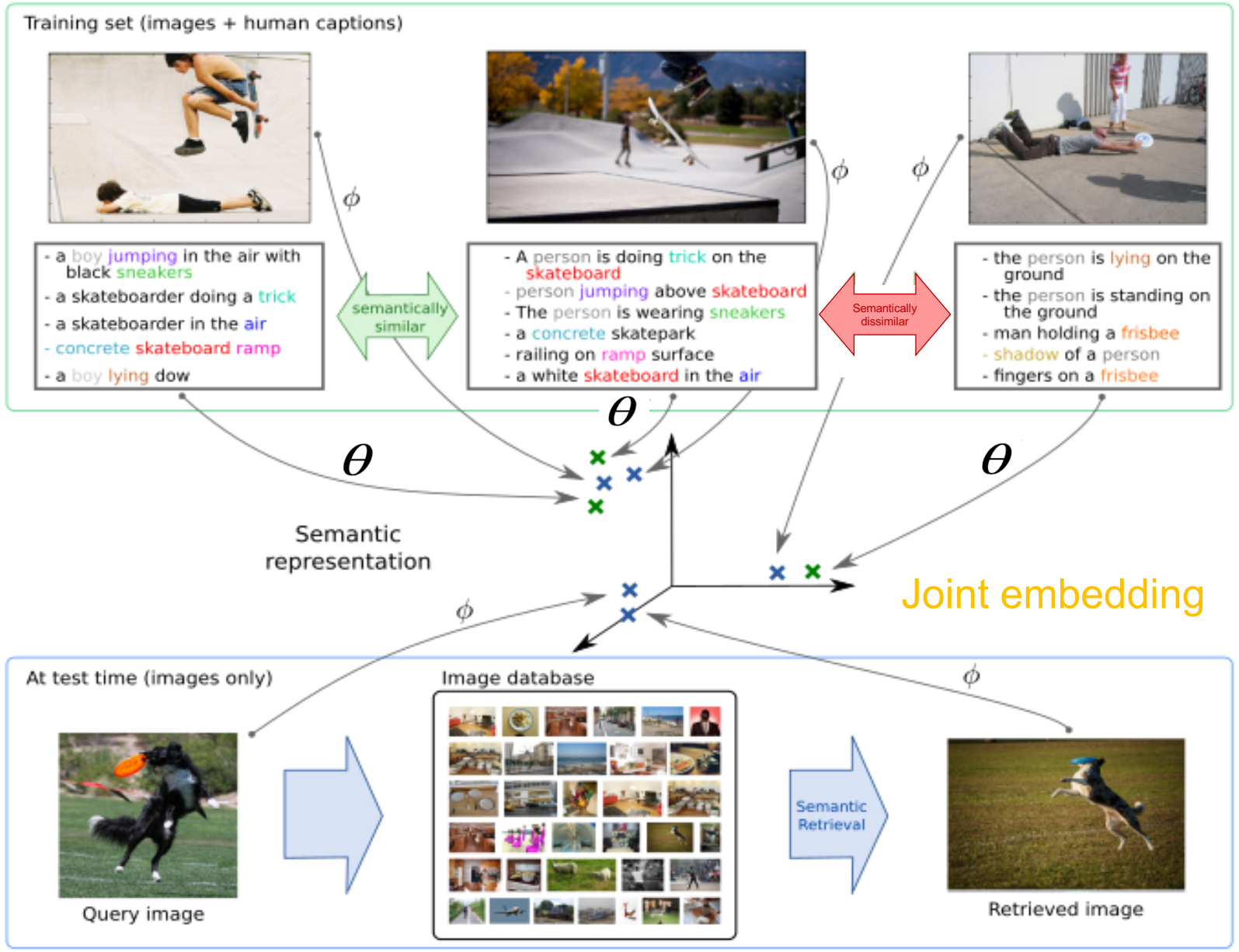
[*Gordo & Larlus. CVPR17*]

Semantic image retrieval

Training the model

The visual representation of images with similar captions are close in the **semantic embedding space**

The textual representation of corresponding captions are close in the **semantic embedding space**

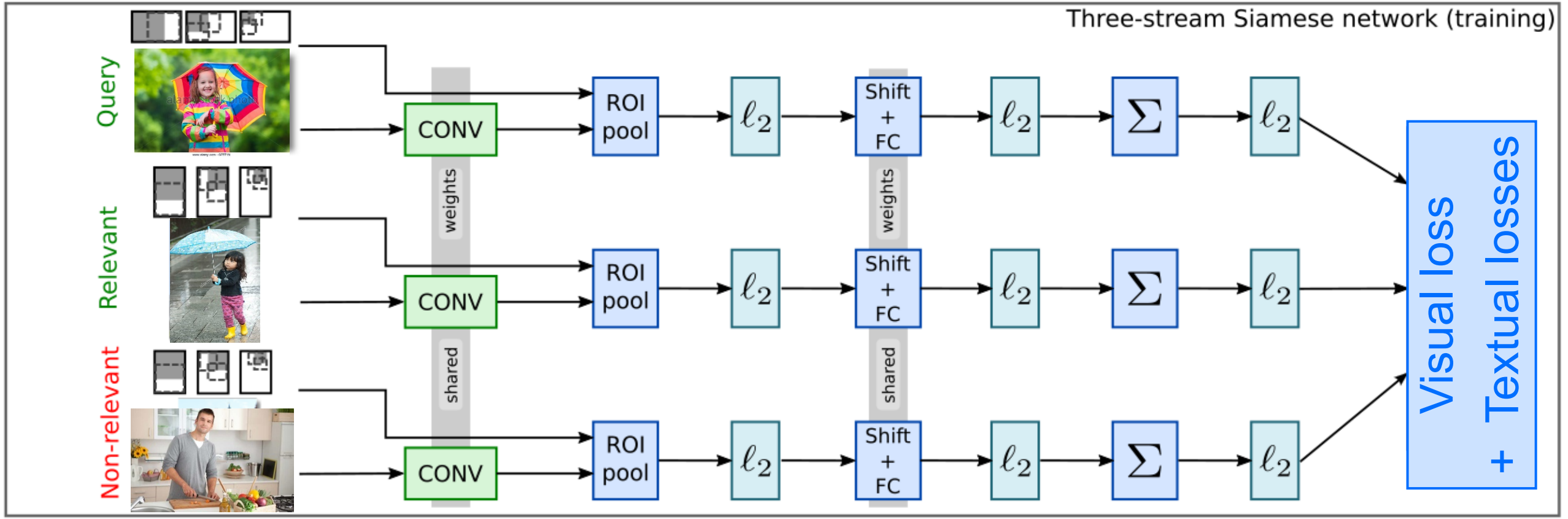


[*Gordo & Larlus. CVPR17*]

Learning a semantic embedding

[Gordo@ECCV16, Gordo@IJCV17]

- Learning to rank
  - Multiple losses are used to learn a good visual representation



Learning a semantic embedding

Visual loss:

$$L_v(q, d^+, d^-) = \frac{1}{2}\max(0, m - \phi_q^T\phi_+ + \phi_q^T\phi_-)$$

Textual losses:

$$L_{t1}(q, d^+, d^-) = \frac{1}{2}\max(0, m - \phi_q^T\theta_+ + \phi_q^T\theta_-)$$

$$L_{t2}(q, d^+, d^-) = \frac{1}{2}\max(0, m - \theta_q^T\phi_+ + \theta_q^T\phi_-)$$

[*Gordo & Larlus. CVPR17*]

Semantic retrieval: qualitative results

[*Gordo & Larlus. CVPR17*]

NAVER LABS EUROPE

Semantic retrieval: qualitative results

[*Gordo & Larlus. CVPR17*]

NAVER LABS EUROPE

Semantic retrieval: leveraging the joint embedding

The joint embedding allows for multimodal queries

[Gordo & Larlus. CVPR17]

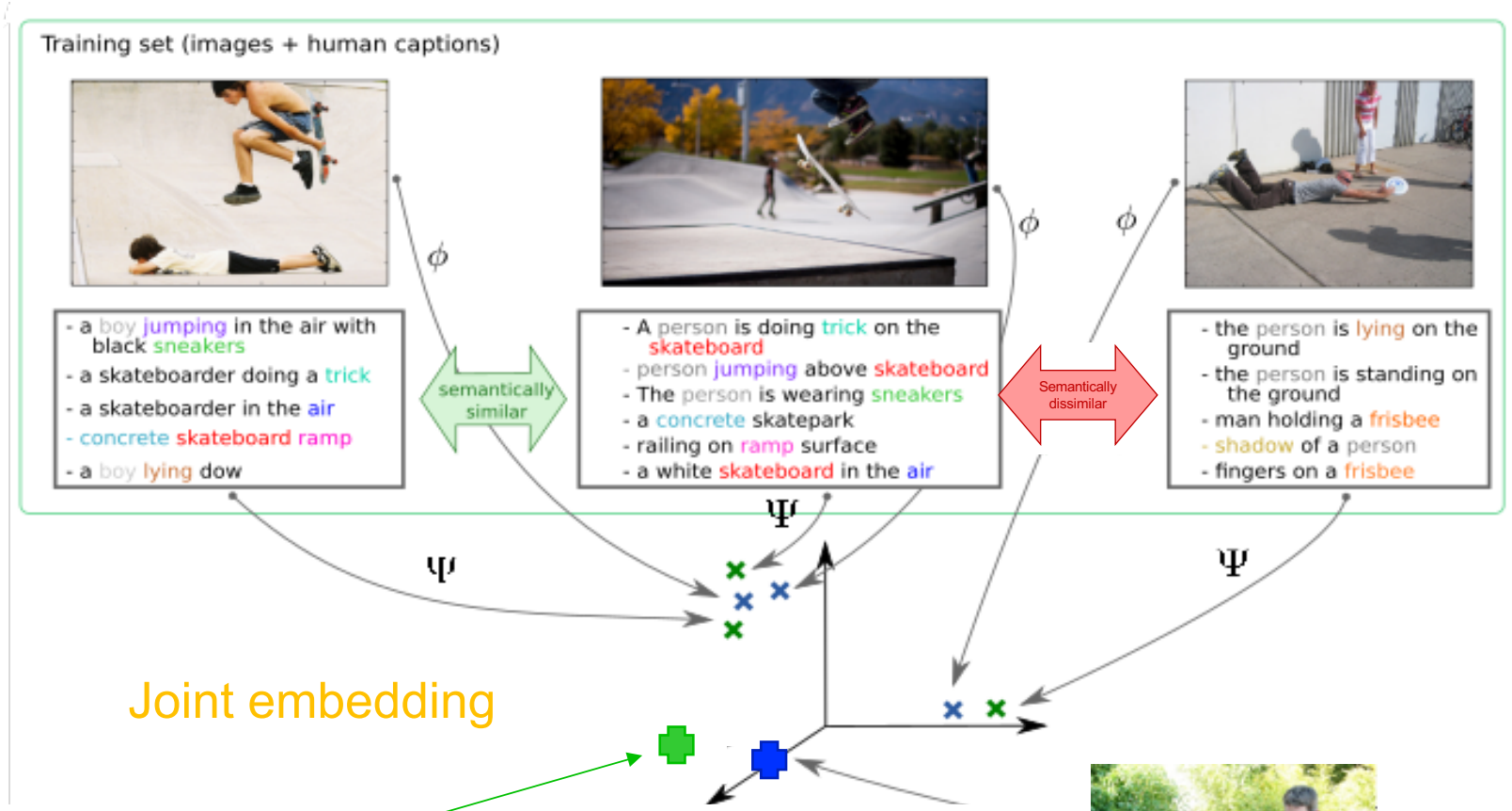
NAVER LABS EUROPE

Text-to-image and Image-to-text retrieval

Offline: build a joint text+image representation by training a **joint embedding space = the semantic embedding**

The semantic embedding space is good for cross-modal searches

Online: embed the query in the joint embedding space, and compare the obtained vector with the embedding vectors of all the elements in the data set



A girl playing with her father

BACKUP SLIDES

Comprendre les données visuelles à grande échelle

Cours 4: recherche d’images, 14 novembre 2019

NAVER LABS EUROPE

Mesures et évaluation – Backup slides

Rappels sur les distances et les mesures de similarité

(si c’est un peu trop loin)

Distances et mesures de similarité : objectif

- Avoir un outil quantitatif pour répondre à la question

  Est-ce que deux entités X et Y se ressemblent ?

- Lorsqu'on désire comparer des entités, on cherche à obtenir un scalaire indiquant la proximité de ces entités

- La mesure utilisée répond à un objectif particulier

- Par exemple:
  - compression d'image : comparer la qualité de reconstruction d'une image compressée avec l'image originale
  - mise en correspondance d'images : indiquer la similarité du contenu de deux images (exemple du début de chapitre)

Distance

- Une **distance** d sur un ensemble *E* est une application de *E* x *E* dans R^+ vérifiant les axiomes suivants

  - (P1) séparation $d(x,y) = 0 \Leftrightarrow x = y$
  - (P2) symétrie $d(x,y) = d(y,x)$
  - (P3) inégalité triangulaire $d(x,z) \leq d(x,y) + d(y,z)$

Distances usuelles sur R^n

$x = (x_1,\ldots,x_i,\ldots,x_n)$ dans R^n

- Distance Euclidienne (ou distance L2)

$$d(x,y) = \sqrt{\sum_i (x_i - y_i)^2}$$

- Distance de Manhattan (ou distance L1)

$$d(x,y) = \sum_i |x_i - y_i|$$

- Plus généralement, distance de Minkowski (ou p-distance)

$$d(x,y) = \sqrt[p]{\sum_i (x_i - y_i)^p}$$

- Cas particulier : distance ∞

$$d(x,y) = \max_i |x_i - y_i|$$

Distance de Mahalanobis

- Observation : les différentes composantes d'un vecteur ne sont pas forcément homogènes, et peuvent être corrélées
- Comment comparer deux vecteurs ?
  - Nécessité de pondérer les composantes
  - Connaissance a priori sur la répartition des points :
    - Matrice de covariance Σ (apprise sur un jeu de données)

Définition : **la distance de Mahalanobis** est

$$d(x,y) = \sqrt{(x-y)^{\mathrm{T}} \Sigma^{-1} (x-y)}$$

- Si $\Sigma = Id$, alors elle est équivalente à la distance Euclidienne
- Si changement de repère $x \rightarrow Lx$ où L est la décomposition de Cholesky de $\Sigma^{-1} = L^{\mathrm{T}}L$ alors la distance de Mahalanobis dans l'espace d'origine = distance L2 dans le repère transformé

Metric learning

Et si on dispose de données annotées avec des exemples de ce qui est similaire et de ce qui ne l'est pas ?

- Utiliser les données pour calculer une bonne métrique

- On peut voir l'apprentissage de cette métrique comme l'apprentissage d'une nouvelle représentation

Metric learning - un exemple

- Exemple dans le domaine linéaire

$$\| x - x' \|_W^2 = (x - x')^\top W^\top W (x - x')$$

- Supervisé:
  - les points appartiennent à des classes (= ils ont des labels)
  - Objectif
    - maximiser la distance entre points de classes différentes
    - minimiser la distance entre points de la même classe
- Trouver W (méthode LMNN)
  - échantillonner des triplets (q, p, n), minimiser $L = \sum_{q=1}^{N} \sum_{p \in P_q} \sum_{n \in N_q} L_{qpn},$
  - descente de gradient en fonction de W $L_{qpn} = \left[1 + \| x_q - x_p \|_W^2 - \| x_q - x_n \|_W^2 \right]_+,$
- Plus pertinent que Mahalanobis
  - proche de l'objectif: classification par plus proche voisin

Metric Learning for Large Scale Image Classification: Generalizing to New Classes at Near-Zero Cost, Thomas Mensink ; Jakob Verbeek ; Florent Perronnin; Gabriela Csurka, ECCV 2012

Autres distances

- Distance du χ^2 (khi-deux ou *chi-square distance*) pour comparer deux distributions (histogrammes)

$$d(x,y) = \sqrt{\sum_i \frac{(x_i - y_i)^2}{x_i + y_i}}$$

- Valorise les variations dans les petites composantes d'un histogramme
- Mahalanobis « du pauvre » quand on n'a pas de données pour calculer une matrice de covariance

Quasi-distance, similarité / dissimilarité

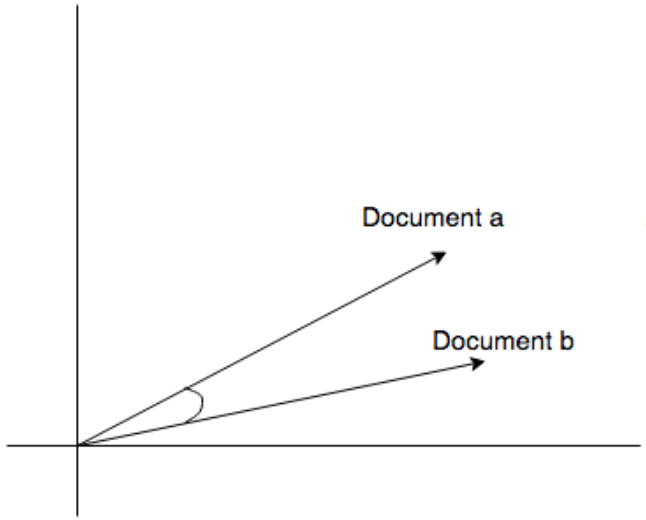
- La notion de distance n'est pas toujours adaptée, car elle impose des axiomes très forts qui ne servent pas directement l'objectif recherché
- Une quasi-distance *q* est une application de *E* x *E* dans R^+ vérifiant les axiomes suivants
  - (P1') x = y **implique** d(x,y) = 0
  - (P2) symétrie : d(x,y) = d(y,x)
  - (P3) inégalité triangulaire: $d(x,z) \leq d(x,y) + d(y,z)$
- Une quasi-distance peut être nulle entre des objets différents
- Plus général encore: mesure de dissimilarité ou de similarité
- Lien évident entre similarité / dissimilarité
  - Faible mesure de similarité = forte mesure de dissimilarité
- Toute distance ou quasi-distance est une mesure de dissimilarité

Exemple

- Le cosinus est une mesure de similarité
  - pour des vecteurs normalisés, équivalent au **produit scalaire**
  - Calcule la similarité entre deux vecteurs en déterminant le cosinus de l'angle entre eux

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Cosine Similarity



$$sim(a,b) = cos\theta = \frac{\vec{a}.\vec{b}}{\| \vec{a} \| \| \vec{b} \|}$$

Merci