Comprendre les données visuelles à grande échelle

ENSIMAG
2019-2020

Karteek Alahari & Diane Larlus

12 décembre 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

Cours 7: Pooling ou « mise en commun »

Comprendre les données visuelles à grande échelle
14 novembre 2019

Représentation d’images par agrégation de représentations locales

Comprendre les données visuelles à grande échelle

Cours 7: pooling, 12 décembre 2019

NAVER LABS EUROPE

Rappel cours 3 !

Description globale

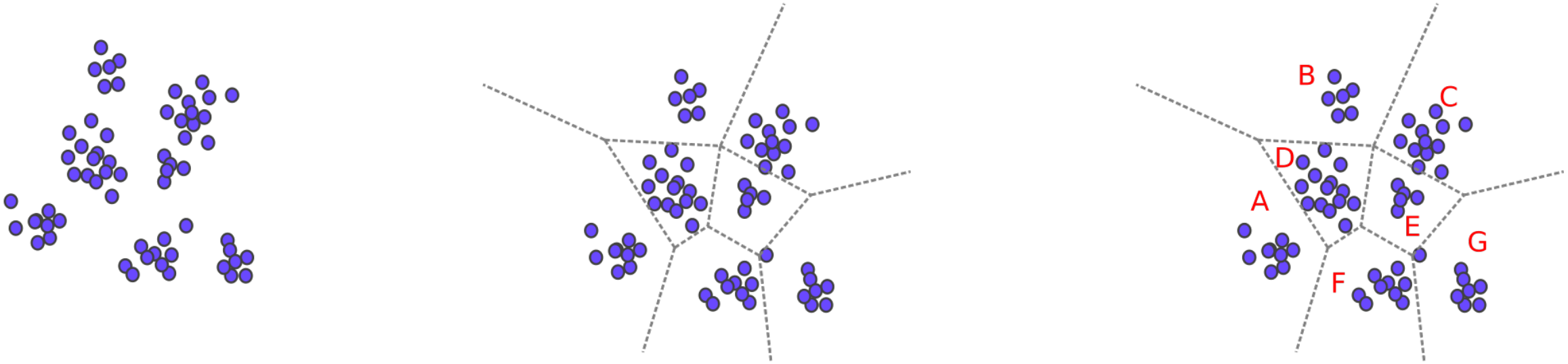
- **Une description globale** est une représentation de l'image dans son ensemble, sous la forme d'un vecteur de taille fixe
- Caractéristiques
  - **un** vecteur de description par objet visuel
  - mesure de (dis-)similarité définie sur l'espace de ces descripteurs

Rappel cours 1 !

Quantization: principle

Principle:

- Discretize the local descriptor space, e.g. SIFT desriptors
- 2 descriptors match i.i.f. they fall in the same bin
- this creates a visual codebook, similar to the textual codebook seen before



NAVER LABS EUROPE

Rappel cours 2 !

Agrégation de descripteurs locaux

- Définir un descripteur global à partir de l'ensemble des descripteurs locaux d'une image
  - Compact, et plus facile à manipuler que les descripteurs locaux de départ
- Contraintes
  - Des images similaires doivent avoir des représentations similaires
  - Des images dissimilaires doivent avoir des représentations dissimilaires
- Compromis
  - Robustes aux transformations (échelle, occultation, éclairage, etc.)
  - Informatifs (bonne description du contenu)
  - Efficaces à calculer, à stocker, à manipuler

Rappel cours 2 !

Agrégation de descripteurs locaux

- La base: « *bag-of-features* », « *bag-of-patches* »
  - *bag* = on perd l'ordre, la géométrie.
    On utilise des ensembles non-ordonnés de descripteurs locaux.

- Quantification: représentation par sac-de-mots,
  ou *bag-of-words* (BoW, aussi appelée *bag-of-visual-words* ou BoV)
  - On suppose une transformation : descripteur local -> index entier
    (par un algorithme de quantification vectorielle = *clustering*)
  - Cette transformation peut être vue comme la création d'un vocabulaire visuel
  - Histogramme de ces entiers = descripteur global

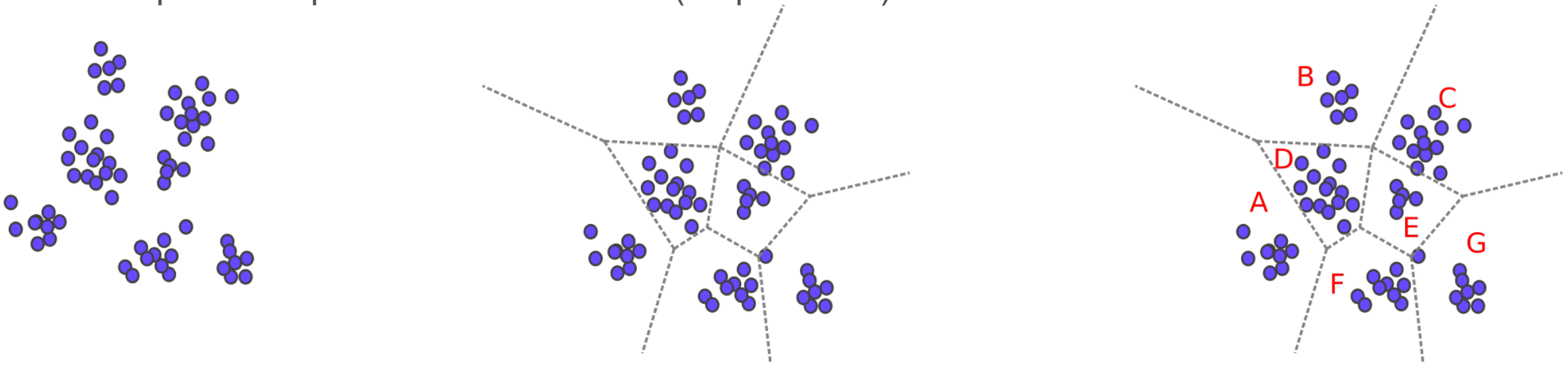
Rappel cours 2 !

Agrégation de descripteurs locaux

Utilisation d'un vocabulaire visuel

Etapes:

- Discrétisation de l'espace des descripteurs, par exemple avec un algorithme de *clustering*
- Chaque descripteur est associé à un (ou plusieurs) mot visuel



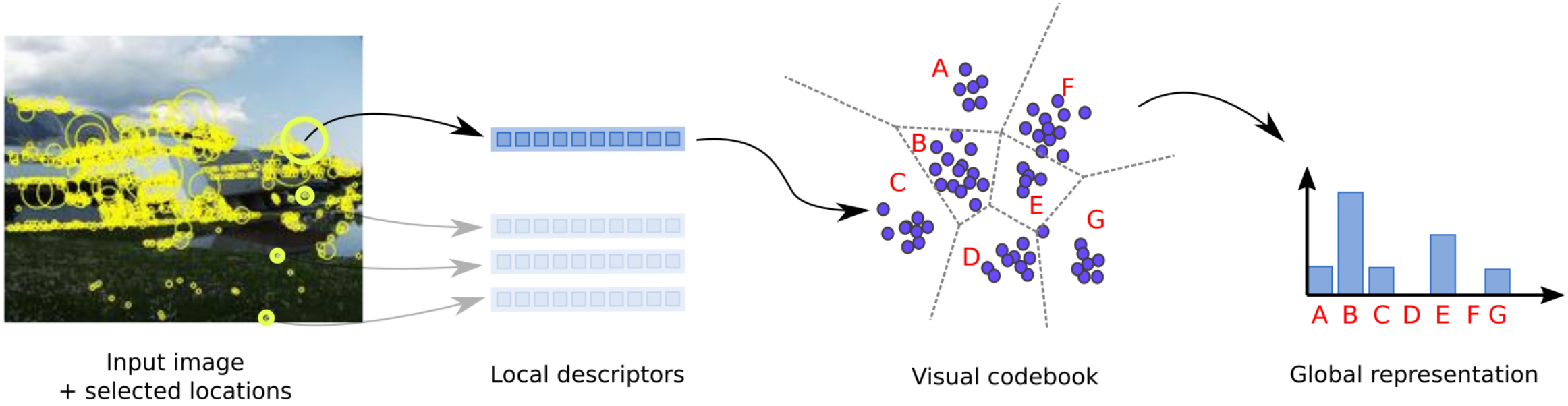
Rappel cours 2 !

From quantization to bag-of-visual-features

Principle

[Sivic & Zisserman. ICCV 2003]
[Csurka et al. ECCV SLCV 2004]

- Extract local descriptors
- Convert local descriptors into visual words, using a visual codebook
- Represent images as a histogram of occurrences



NAVER LABS EUROPE

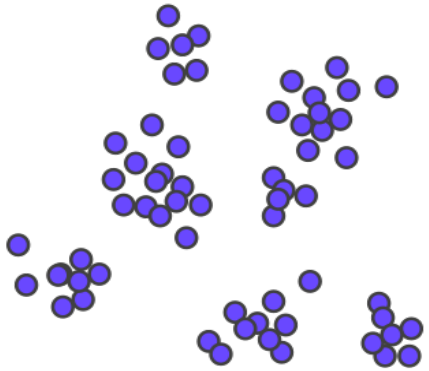
How can we refine this description?

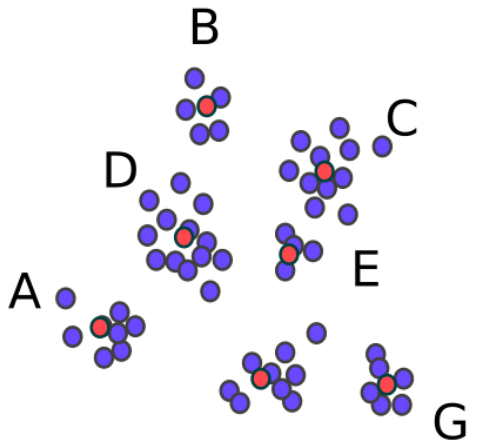
Relatively coarse representation

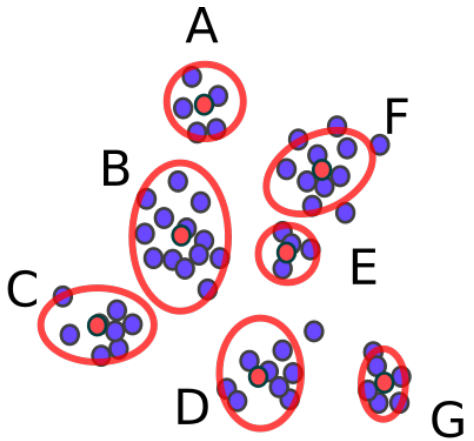
- Solution 1: more entries in the codebook
  - drawback: significant computational cost

[Li et al. ICCV 2009]
[Yang et al. ECCV 2010]

- Solution 2: beyond counting, adding higher order statistics
  - Mean: VLAD
  - Variance: Fisher Vector





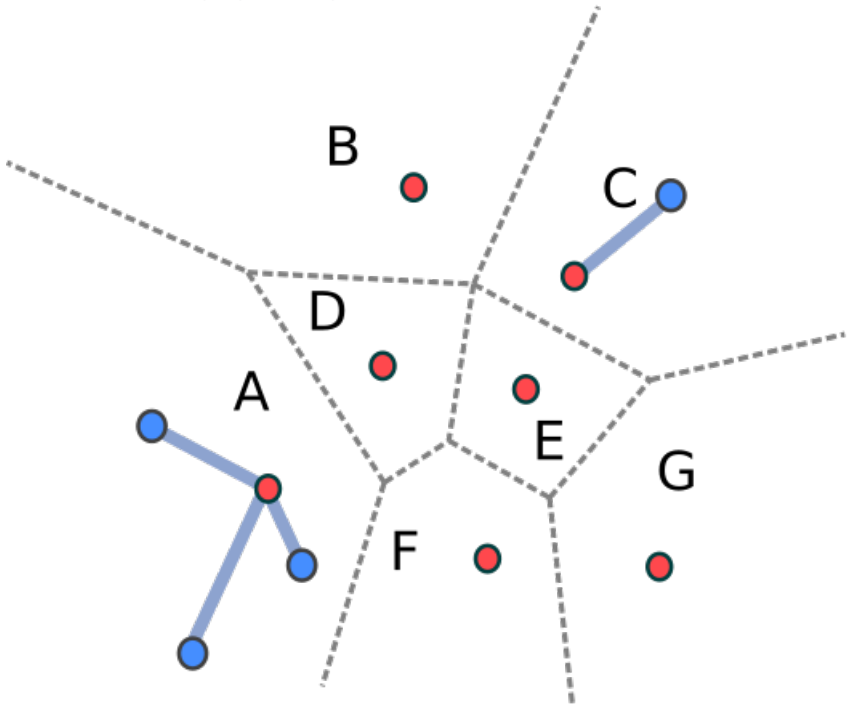


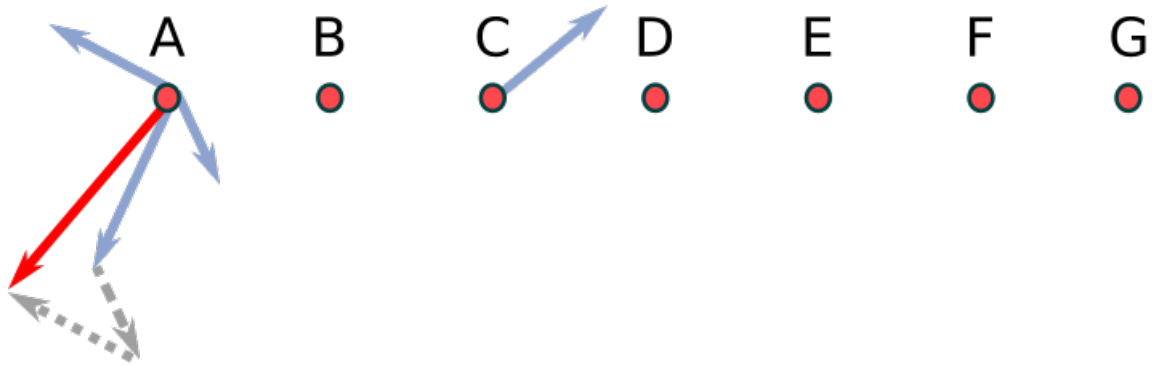
NAVER LABS EUROPE

Extension to higher order statistics

Mean: VLAD (Vector of Locally Aggregated Descriptors)

- Aggregate all descriptors assigned to the same visual word



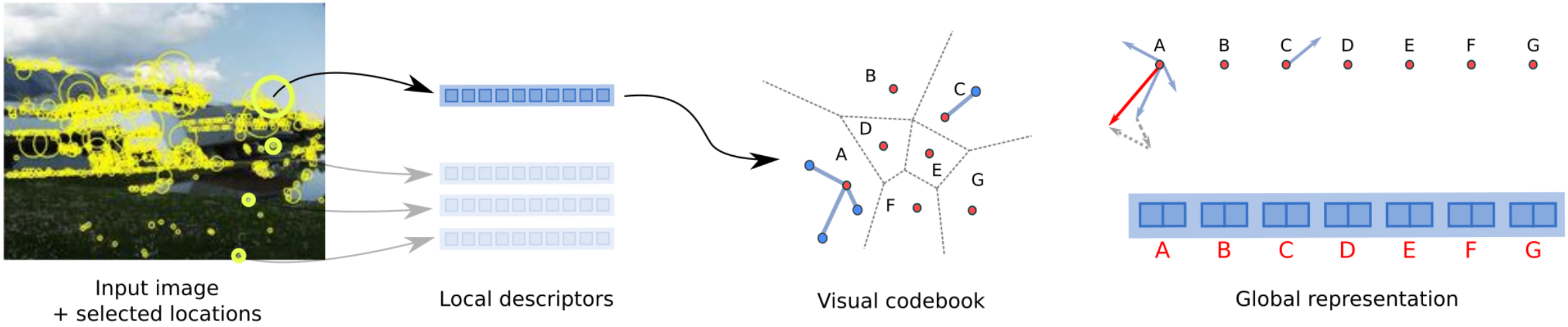


[Jégou et al. CVPR 2010]

Extension to higher order statistics

Mean: VLAD (Vector of Locally Aggregated Descriptors)

- Aggregate all descriptors assigned to the same visual word
- Concatenate vectors for individual words



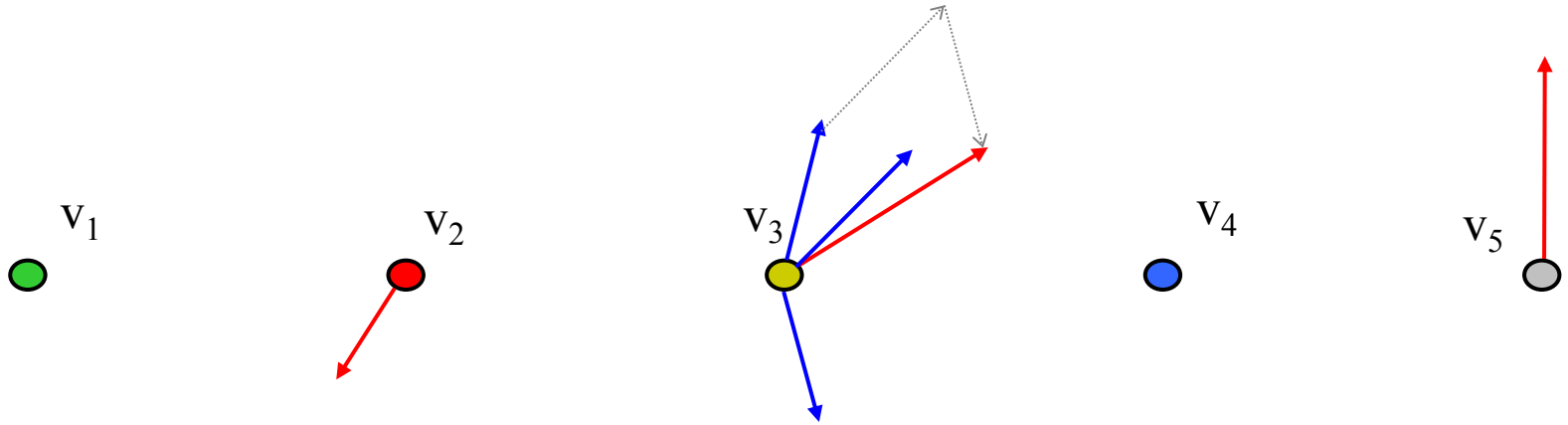
[Jégou et al. CVPR 2010]

NAVER LABS EUROPE

VLAD: Vector of Locally Aggregated Descriptors

En pratique:

- Espace des descripteurs à D-dimensions (SIFT: D=128)
- *k* centres : $c_1,\ldots,c_k$



- Sortie: $v_1 \ldots v_k$ = descripteurs de taille k*D
- L2-normalisation
- Typiquement *k* = 16 or 64 : descripteur de 2048 or 8192 D
- Mesure de similarité = distance L2

VLAD: Vector of Locally Aggregated Descriptors

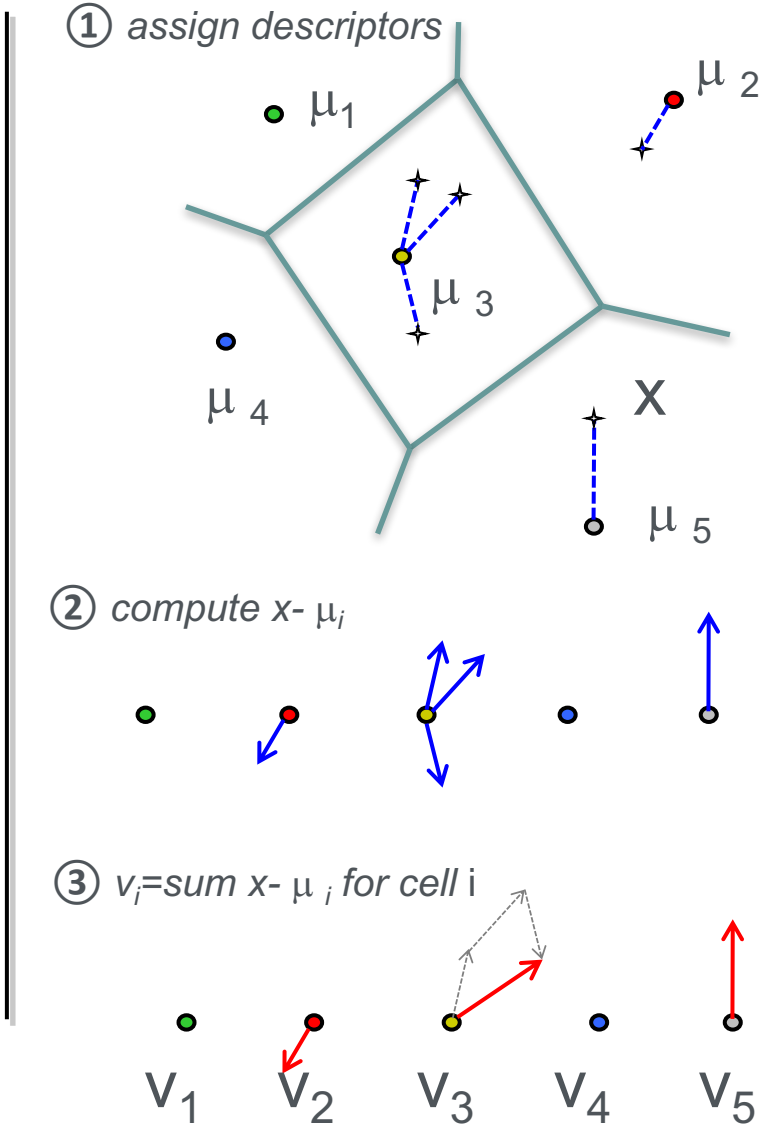
Offline:
apprendre le vocabulaire visuel $\{\mu_1, \ldots, \mu_N\}$,
par exemple avec *K-means*

Online:
Entrée: un ensemble de descripteurs locaux $X = \{x_1, \ldots, x_T\}$:

- ① affectation des descripteurs au mot le plus proche:

$$NN(x_t) = \arg\min_{\mu_i} \|x_t - \mu_i\|$$

- ②③ calcul de $v_i = \sum_{x_t : NN(x_t) = \mu_i} (x_t - \mu_i)$
- Concaténation des v_i's + ℓ_2-normalisation

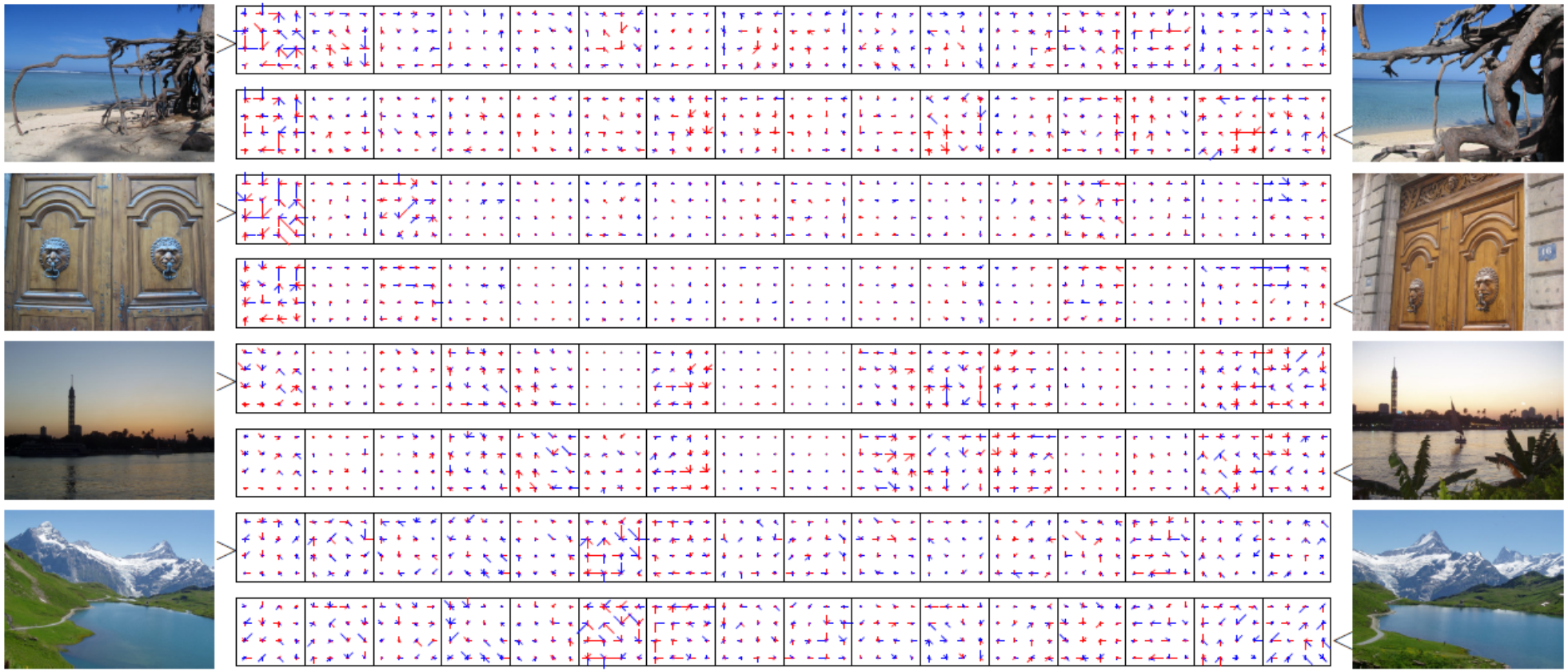


Jégou, Douze, Schmid and Pérez, "Aggregating local descriptors into a compact image representation", CVPR'10.

See also: Zhou, Yu, Zhang and Huang, "Image classification using super-vector coding of local image descriptors", ECCV'10.

VLAD: Vector of Locally Aggregated Descriptors

- Illustration du résultat de l'étape $v_i = \sum_{x_t : NN(x_t)=\mu_i} (x_t - \mu_i)$ dans le cas de descripteurs SIFT

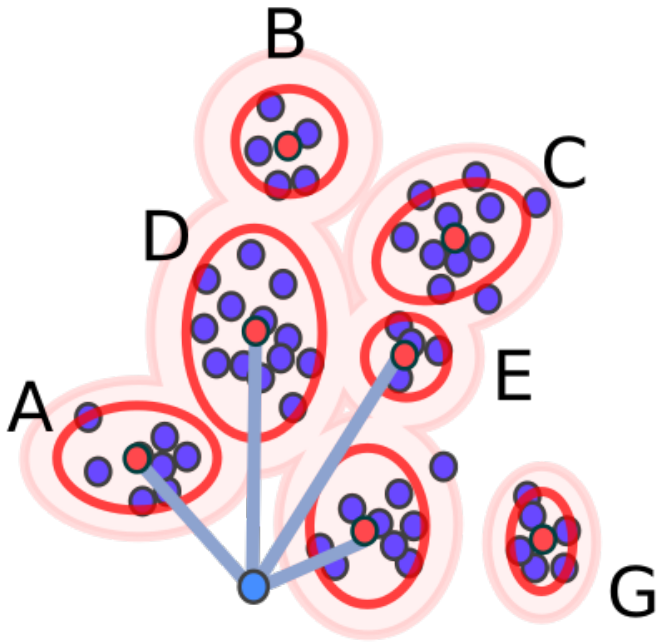


Jégou, Douze, Schmid and Pérez, "Aggregating local descriptors into a compact image representation", CVPR'10.

Extension to higher order statistics

Mean + Variance: Fisher-Vector

- Probabilistic codebook as a mixture of Gaussians
- Descriptors soft-assigned to words
- Compute the gradient of the log-likelihood w.r.t. the parameters of the model



[Perronnin and Dance. CVPR07]
[Perronnin et al. CVPR10]

NAVER LABS EUROPE

Fisher Vector

fonction de score

Principe du vecteur de Fisher

- Etant donné une fonction de vraisemblance (*likelihood function*) u_λ de paramètres λ, la fonction de score d'un échantillon x_t est donnée par :

$$g_\lambda(x_t) = \nabla_\lambda \log u_\lambda(x_t)$$

$\rightarrow$ **dont la dimension dépend du nombre de paramètres**

- Intuition: indique la direction dans laquelle les paramètres λ du modèle devraient être modifiés pour mieux représenter les données

[Slide credit: F. Perronnin, International Computer Vision Summer School. 2014]

Fisher vector

Fisher information matrix

- **Fisher information matrix** (FIM) or negative Hessian:

$$F_\lambda = E_{x \sim u_\lambda}[g_\lambda(x) g_\lambda(x)^T]$$

- Measure similarity between gradient vectors using the **Fisher Kernel (FK)**:

[Jaakkola and Haussler, "Exploiting generative models in discriminative classifiers", NIPS'98]

$$K(x, z) = g_\lambda(x)^T {F_\lambda}^{-1} g_\lambda(z)$$

$\rightarrow$ can be interpreted as a score whitening

- As the FIM is PSD, we can write: ${F_\lambda}^{-1} = {L_\lambda}^T L_\lambda$

and the FK can be rewritten as a dot product between **Fisher Vectors** (FV):

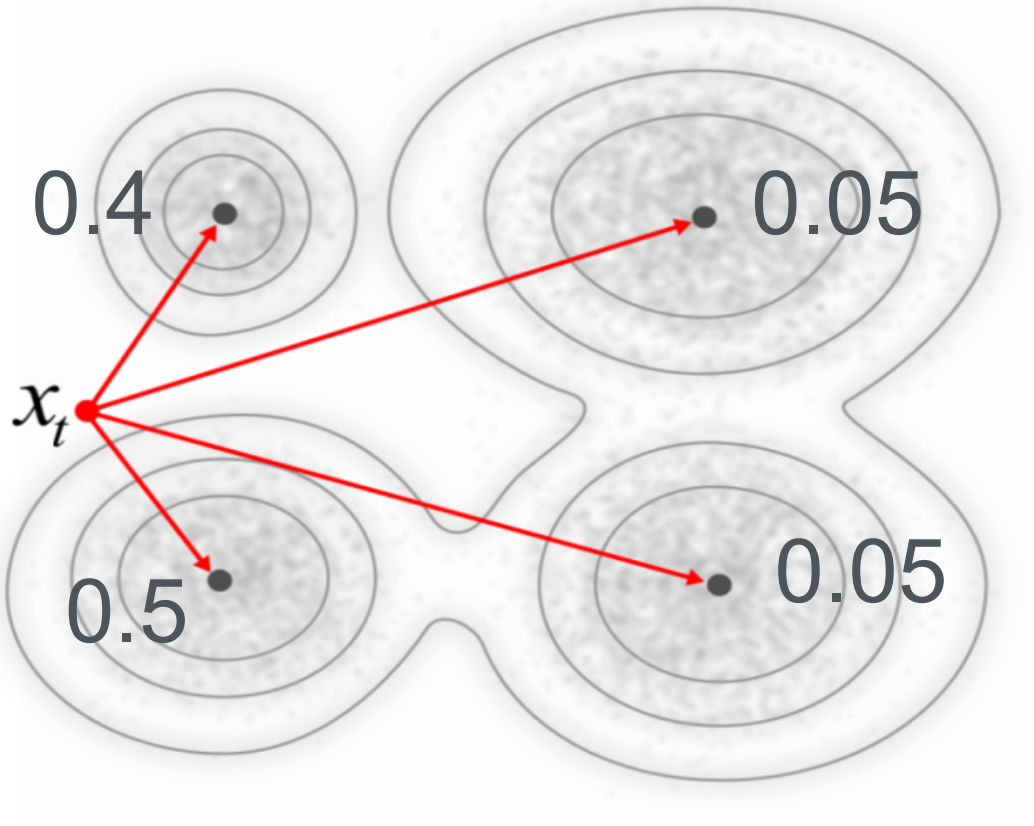
$$\varphi_\lambda^{fv} \quad (x_t) = L_\lambda \, g(x_t)$$

[Slide credit: F. Perronnin, International Computer Vision Summer School. 2014]

Fisher vector

Application aux images

- Dans le cas des images, typiquement u_λ est une mixture de gaussiennes ou *Gaussian Mixture Model* (GMM):
  - $u_\lambda(x) = \sum_{i=1}^{N} w_i u_i(x)$



[Slide credit: F. Perronnin, International Computer Vision Summer School. 2014]

Fisher vector

Application aux images

- Dans le cas des images, typiquement u_λ est une mixture de gaussiennes ou *Gaussian Mixture Model* (GMM):
  - $u_\lambda(x) = \sum_{i=1}^{N} w_i u_i(x)$
  - $u_i(x) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_i)'{\Sigma_i}^{-1}(x-\mu_i)\right\}$

  → ils constituent les **mots visuels**

  et l'ensemble des paramètres est $\lambda = \{w_i, \mu_i, \Sigma_i, i = 1 \dots N\}$

- On suppose généralement que la matrice est diagonale pour des raisons de complexité de calcul

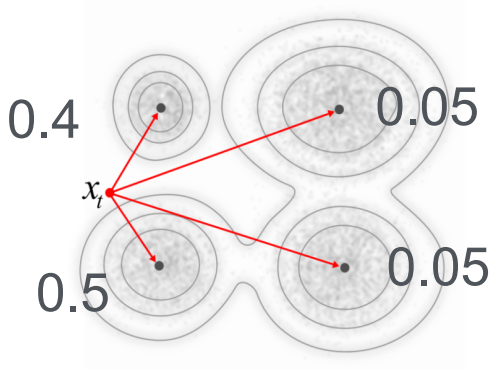
$$\Sigma_i = diag({\sigma_i}^2)$$

→ FV = concaténation des gradients selon les paramètres w_i, μ_i, et σ_i

[Slide credit: F. Perronnin, International Computer Vision Summer School. 2014]

Fisher vector

Résumé



Offline: apprendre un modèle GMM de paramètres $\lambda = \{w_i, \mu_i, \Sigma_i, i = 1 \dots N\}$

Online: étant donné un ensemble de descripteurs $\mathrm{X} = \{x_1, \dots, x_T\}$ pour une image

- Pour chaque x_t:
  - Assignement souple à une gaussienne: $\gamma_t(i) = \frac{w_i u_i(x_t)}{\sum_{k=1}^{N} w_k u_k(x_t)}$
  - Accumulation des valeurs

    $\varphi_\mu \; += \left[\dots, \frac{\gamma_t(i)}{\sigma_i \sqrt{w_i}}(x_t - \mu_i), \dots\right]$

    et $\varphi_\sigma \; += \left[\dots, \frac{\gamma_t(i)}{\sqrt{2w_i}}\left(\frac{(x_t - \mu_i)^2}{{\sigma_i}^2} - 1\right), \dots\right]$
- *power-normalization* + normalisation L2

Voir aussi les implémentations:

- INRIA: http://lear.inrialpes.fr/src/inria_fisher/
- Oxford: http://www.robots.ox.ac.uk/~vgg/software/enceval_toolkit/

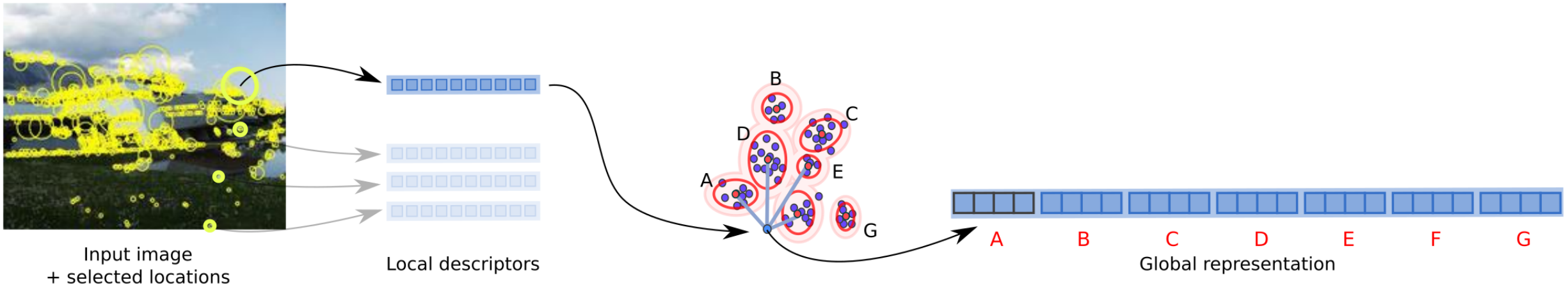
[Slide credit: F. Perronnin, International Computer Vision Summer School. 2014]

Extension to higher order statistics

Mean + Variance: Fisher-Vector

- Aggregate the contribution of all local descriptors
- Concatenate statistics for all the visual words

[Perronnin and Dance. CVPR07]
[Perronnin et al. CVPR10]

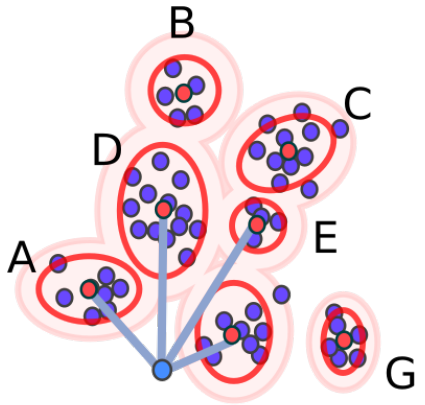


NAVER LABS EUROPE

Fisher vector

En pratique:

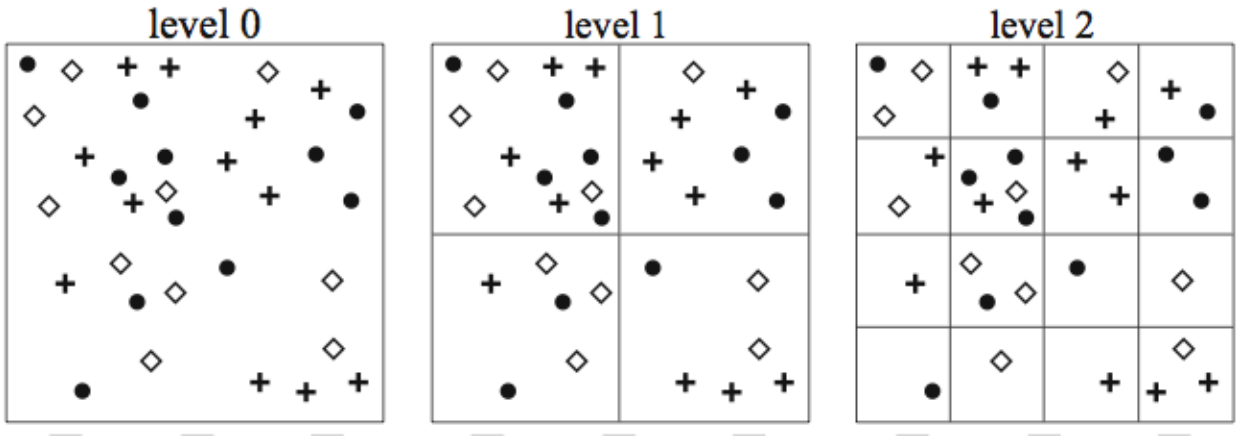
- Espace des descripteurs à d-dimensions (SIFT: d=128)
- Réduction de la dimension par ACP *(PCA)* (e.g. *D*=64)
- *K* gaussiennes de paramètres $\lambda = \{w_i, \mu_i, \Sigma_i, i = 1 \dots K\}$
- Sortie:
  - φ_μ : descripteur de taille *K*D*
  - φ_σ *:* descripteur de taille *K∗D*
- Concaténation et L2-normalisation
- Descripteur final : *2*K*D* dimensions



Comment préserver des informations géométriques ?

Pyramide spatiale

- Problème: les représentations par *« bag-of-patches »* ne gardent aucune information sur la géométrie
- Solution: calculer des agrégats de descripteurs sur des sous-images
  - géométrie rigide
  - taille des vecteurs multipliée (pour l'exemple ci-dessous par 21)



Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories, S. Lazebnik, C. Schmid, J. Ponce, CVPR06

Exercice

- Calculer la représentation *Bag-of-Words*
- Calculer la représentation VLAD

(au tableau)

Représentation d’images par apprentissage profond

Comprendre les données visuelles à grande échelle

Cours 7: pooling, 12 décembre 2019

NAVER LABS EUROPE

Credits

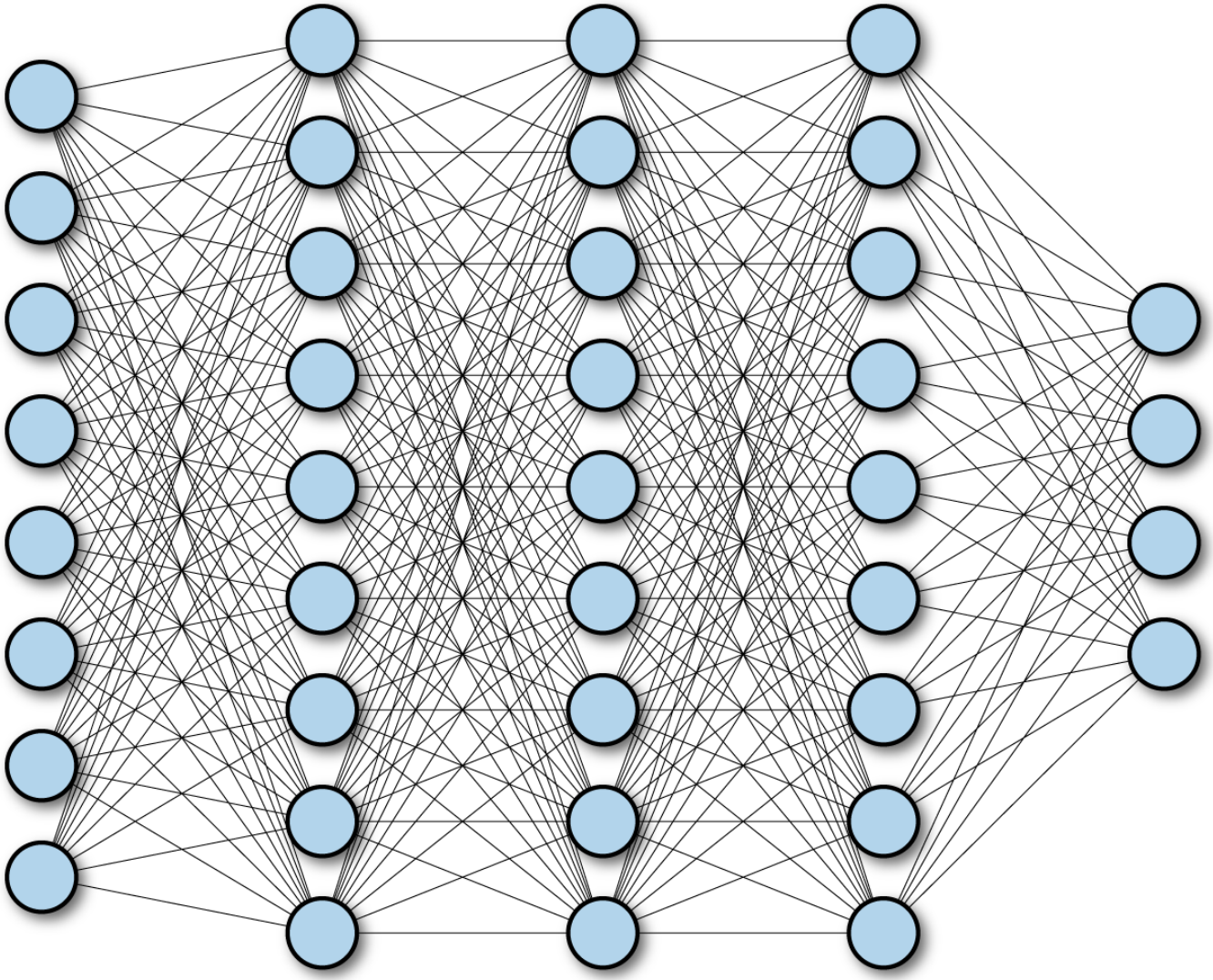
- Some material from

  - Standford **CS231n** course
    “Convolutional Neural Networks for Visual Recognition”
    http://cs231n.stanford.edu/

  - Talk of Martin Görner (Google Cloud)
    “Learn TensorFlow and deep learning, without a Ph.D.”
    https://cloud.google.com/blog/big-data/2017/01/learn-tensorflow-and-deep-learning-without-a-phd

  - Talk of Andrej Karpathy (OpenAI)
    “ Convolutional Neural Networks ”
    https://www.youtube.com/watch?v=u6aEYuemt0M

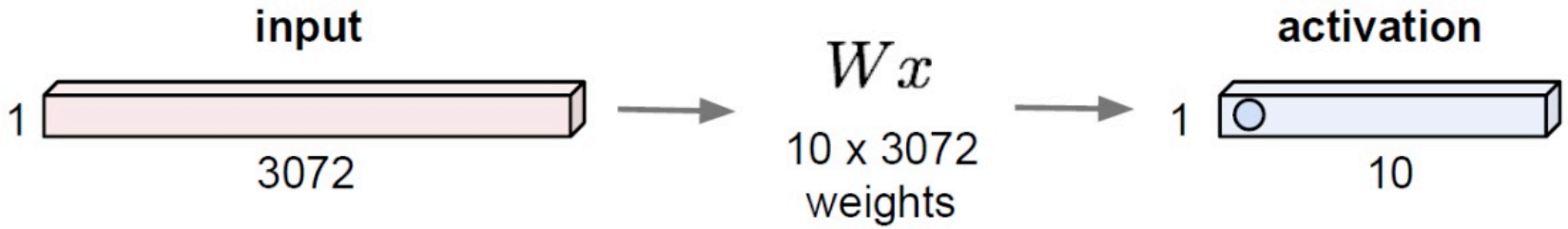
Fully Connected Layer



[Credit: https://www.oreilly.com/library/view/tensorflow-for-deep]

Fully Connected Layer

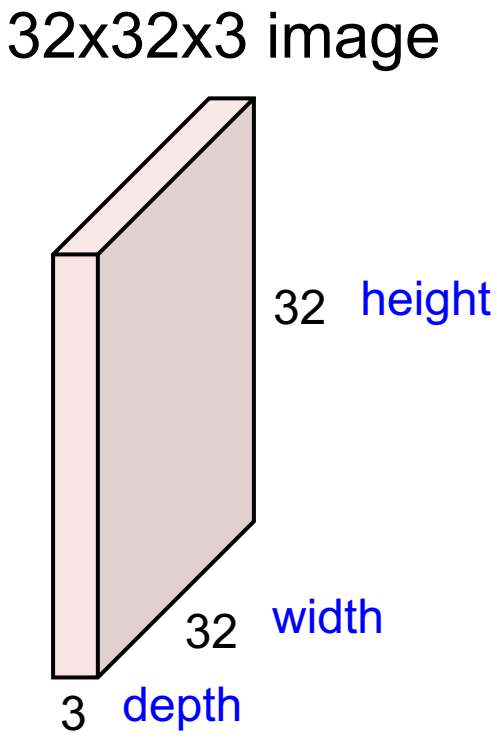
32x32x3 image -> stretch to 3072x1



- What is wrong with fully-connected layers?
  - Fixed-input size
  - No translation invariance
  - Quickly blow-up the number of parameters
  - Loose all the structure of the image
- Solution
  - Convolutional Neural Networks

[Credit: Karpathy - Standford – CS231 course]

Convolution Layer

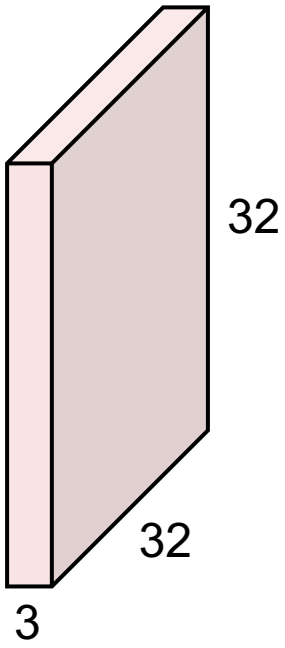


[Credit: Karpathy - Standford – CS231 course]

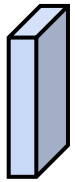
Convolution Layer

Filters always extend the full depth of the input volume

32x32x3 image



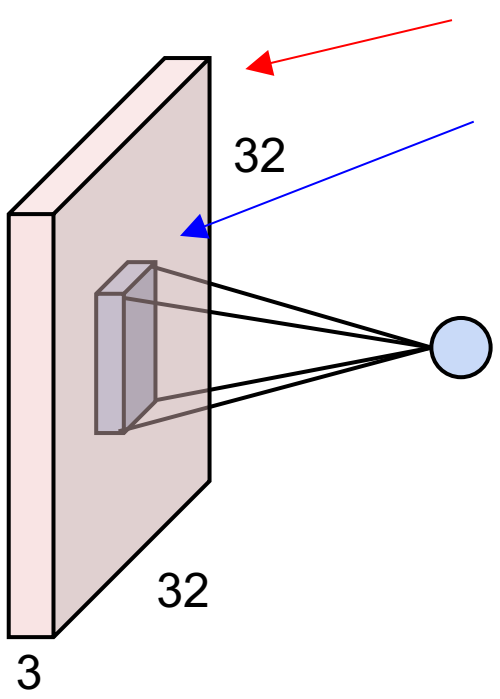
5x5x3 filter



Convolve the filter with the image
i.e. “slide over the image spatially, computing dot products”

[Credit: Karpathy - Standford – CS231 course]

Convolution Layer

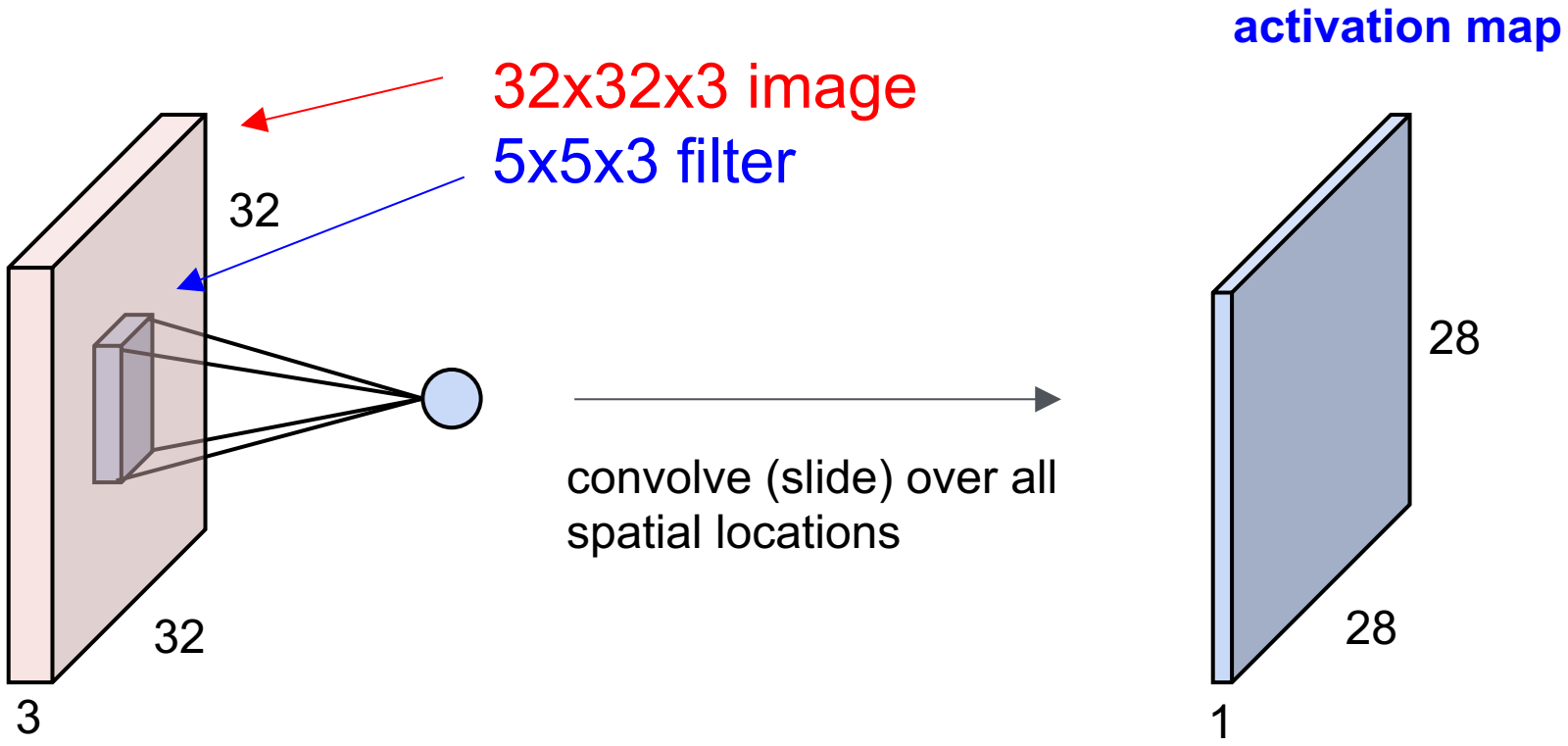


1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image

$$w^T x + b$$

[Credit: Karpathy - Standford – CS231 course]

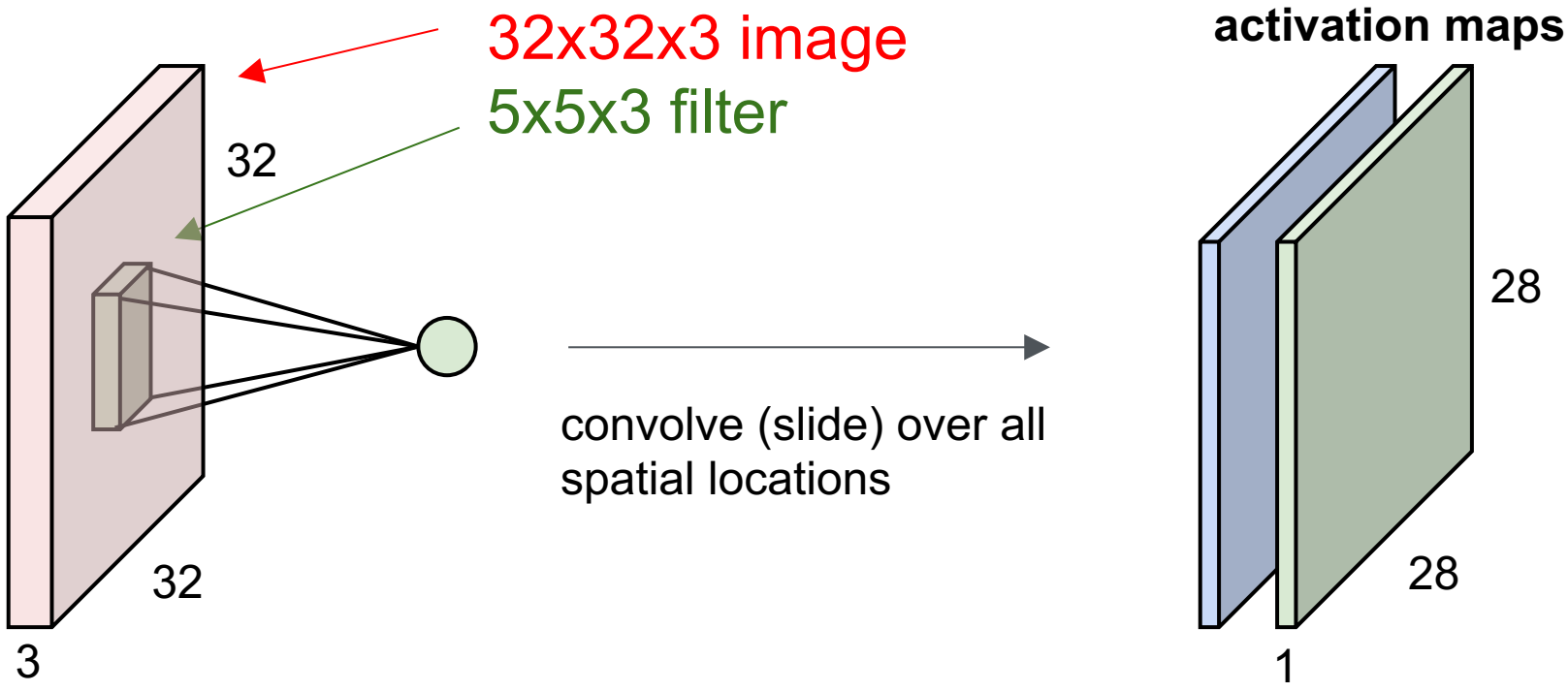
Convolution Layer



[Credit: Karpathy - Standford – CS231 course]

Convolution Layer

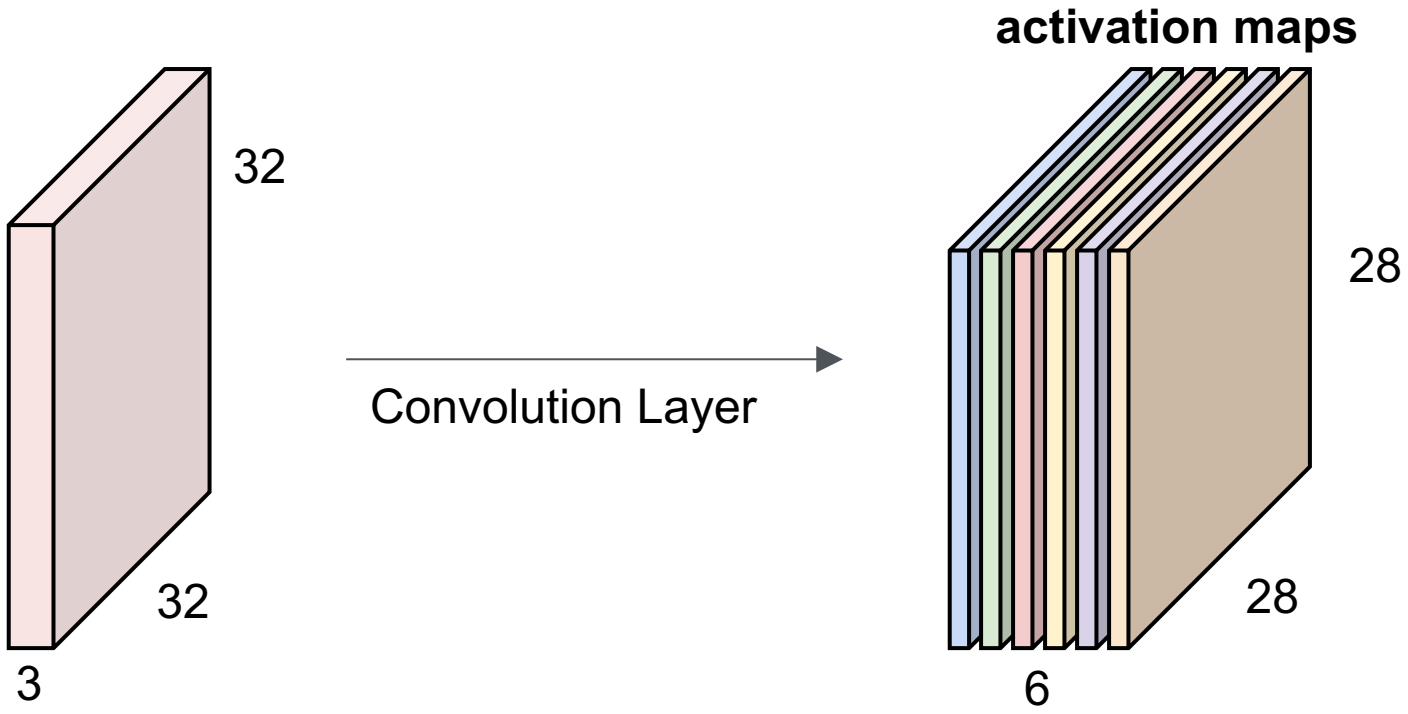
consider a second, green filter



[Credit: Karpathy - Standford – CS231 course]

Convolution Layer

For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:

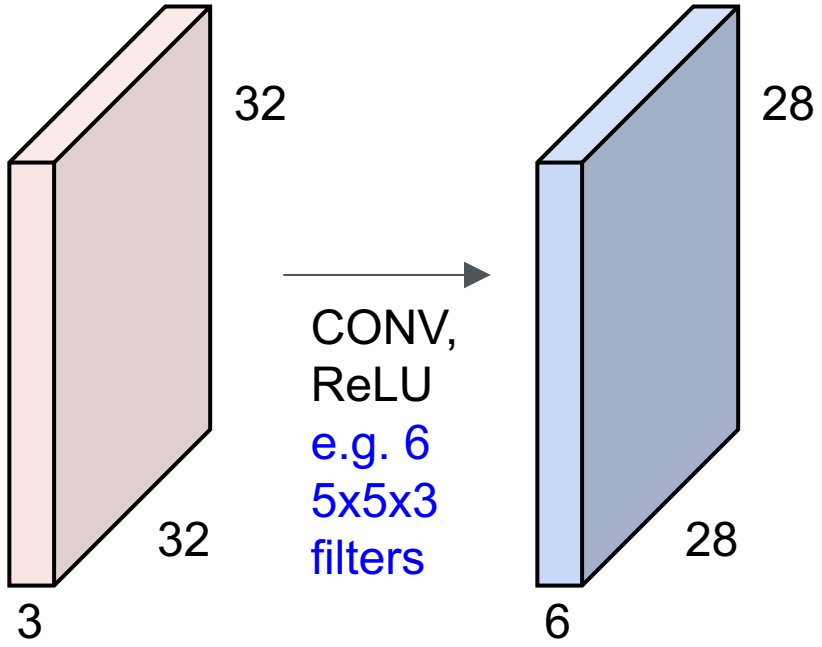


We stack these up to get a "new image" of size 28x28x6!

[Credit: Karpathy - Standford – CS231 course]

Convolution Layer

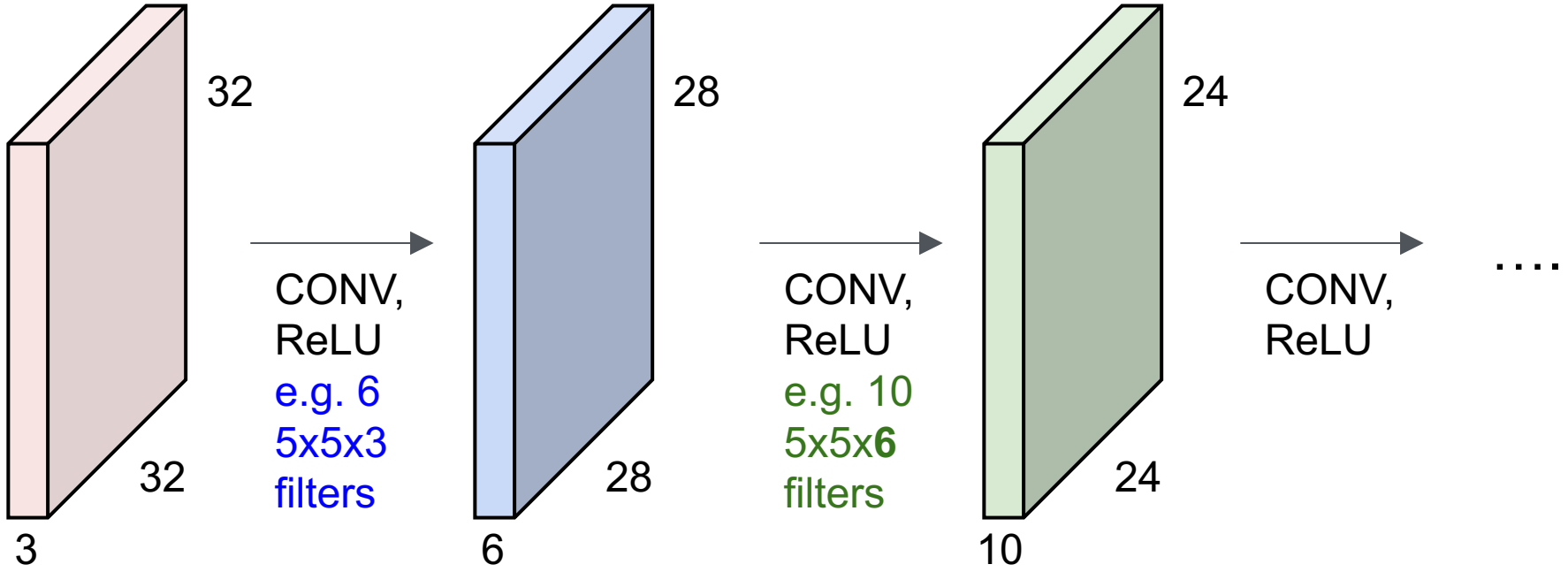
Preview: ConvNet is a sequence of Convolutional Layers, interspersed with activation functions



[Credit: Karpathy - Standford – CS231 course]

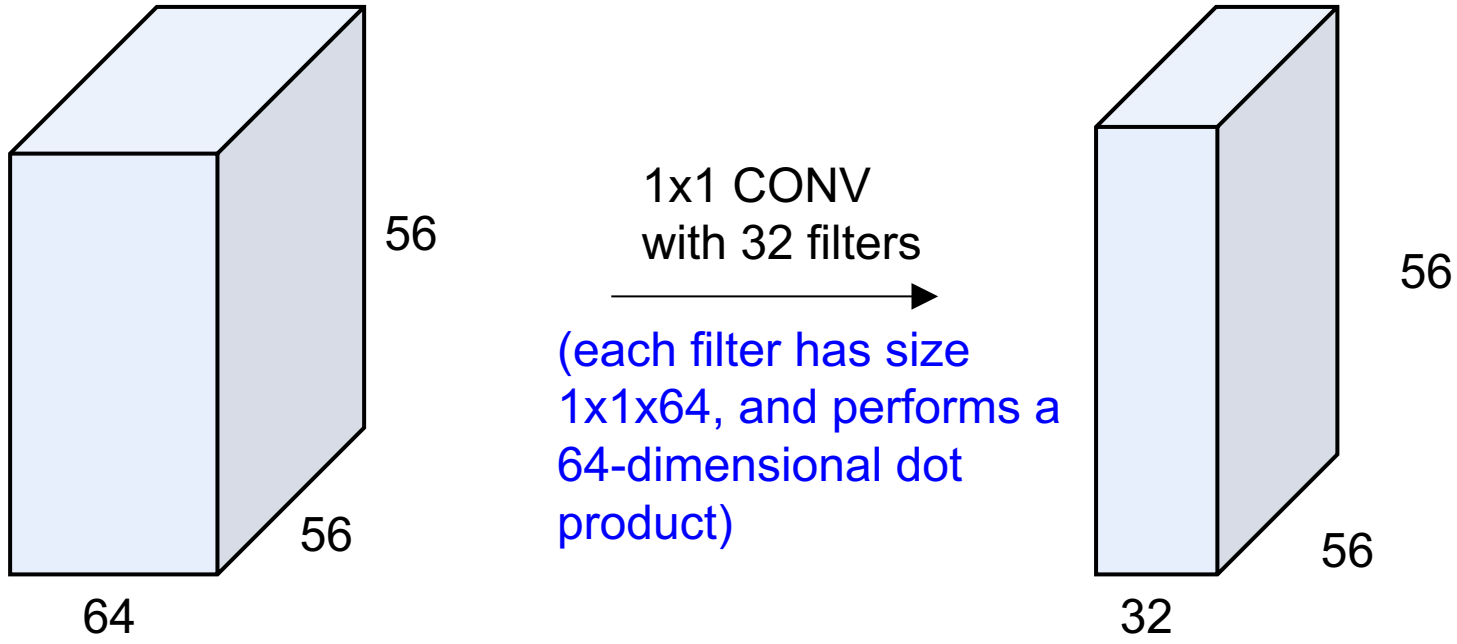
Convolution Layer

Preview: ConvNet is a sequence of Convolutional Layers, interspersed with activation functions



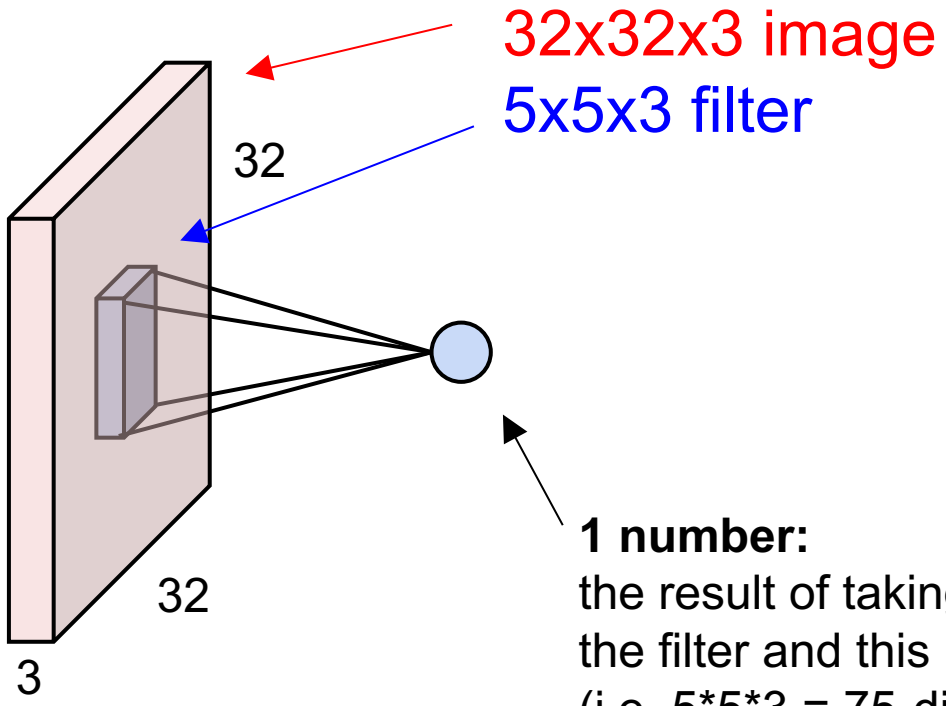
[Credit: Karpathy - Standford – CS231 course]

- Note that a 1x1 convolution layer makes perfect sense



[Credit: Karpathy - Standford – CS231 course]

Analogy with the brain

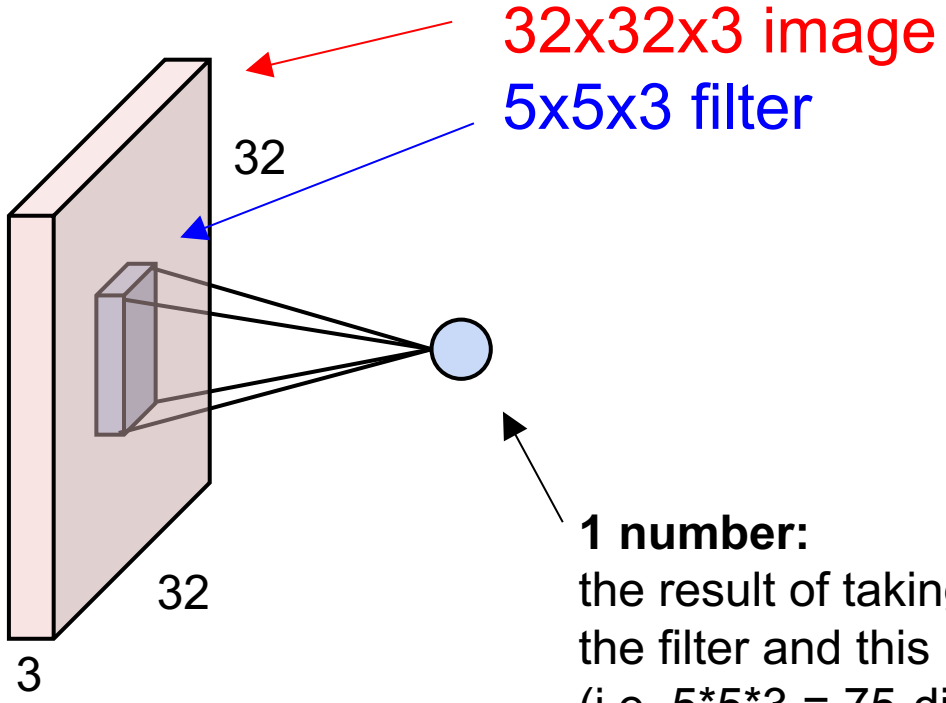


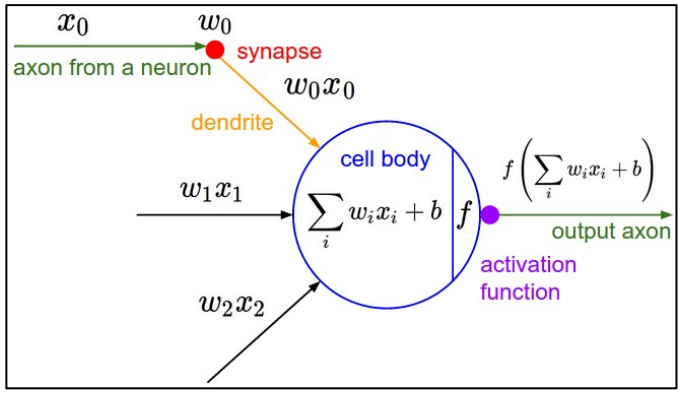
1 number:
the result of taking a dot product between the filter and this part of the image
(i.e. 5*5*3 = 75-dimensional dot product)

[Credit: Karpathy - Standford – CS231 course]

Analogy with the brain

The brain/neuron view of CONV Layer





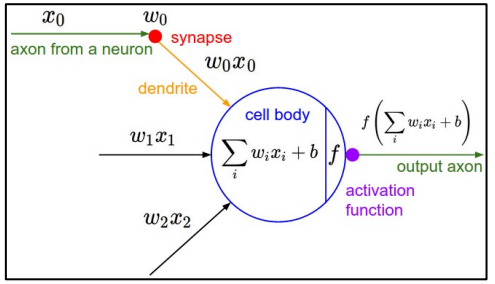
It's just a neuron with local connectivity...

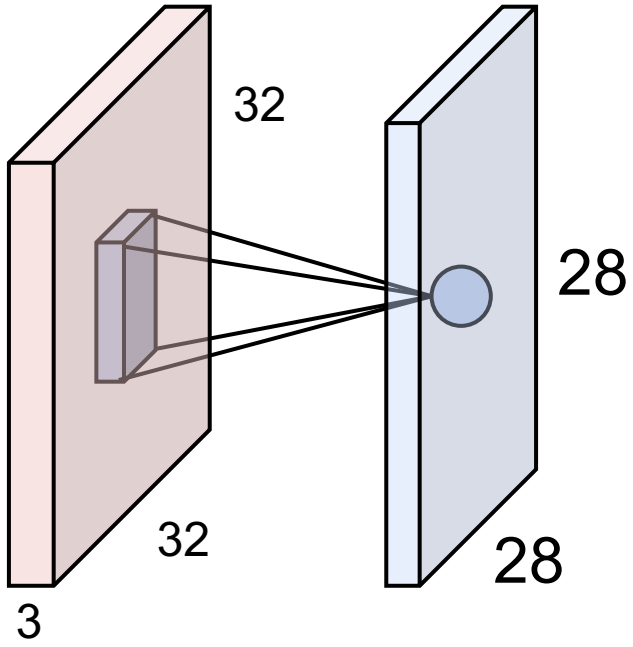
1 number:
the result of taking a dot product between the filter and this part of the image
(i.e. 5*5*3 = 75-dimensional dot product)

[Credit: Karpathy - Standford – CS231 course]

Analogy with the brain

The brain/neuron view of CONV Layer





An activation map is a 28x28 sheet of neuron outputs:

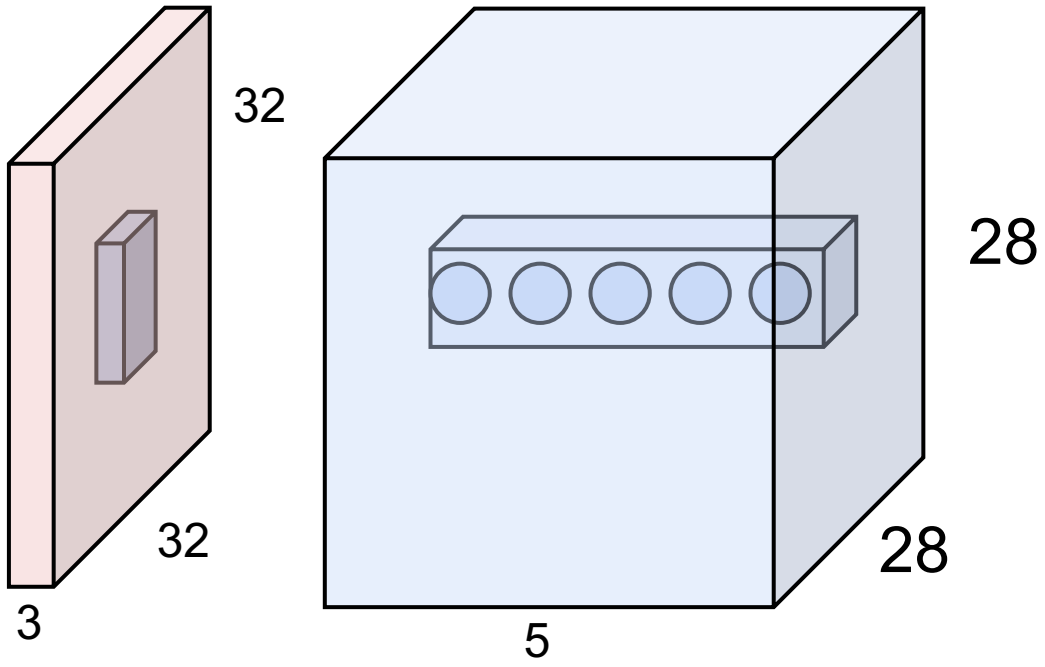
1. Each is connected to a small region in the input
2. All of them share parameters

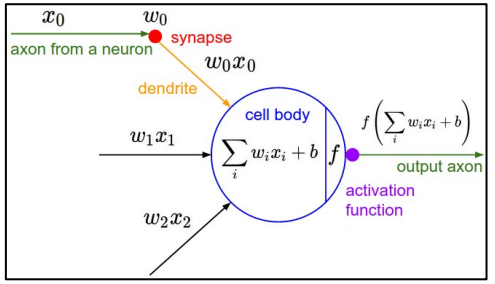
"5x5 filter" -> "5x5 receptive field for each neuron"

[Credit: Karpathy - Standford – CS231 course]

Analogy with the brain

The brain/neuron view of CONV Layer





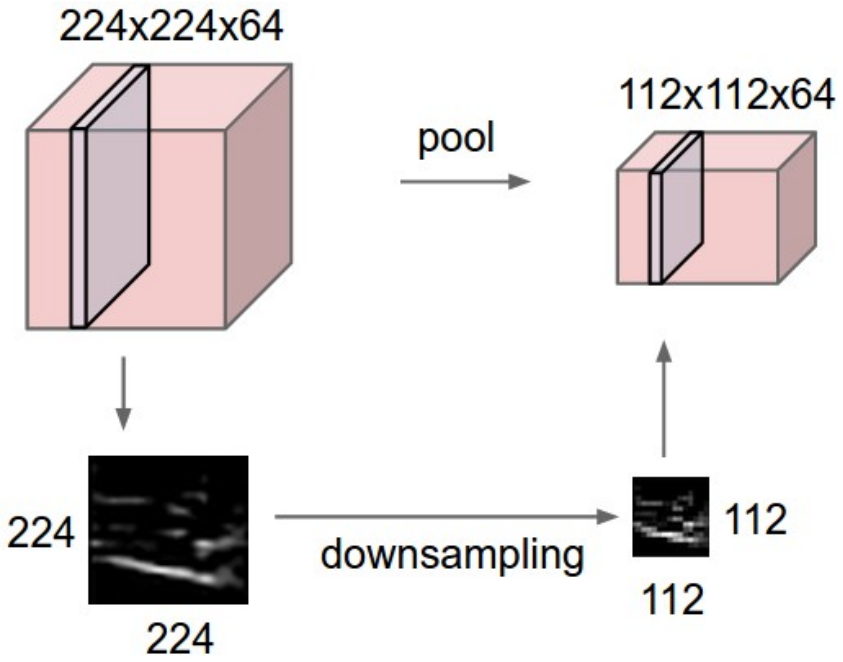
E.g. with 5 filters,
CONV layer consists of neurons arranged in a 3D grid (28x28x5)

There will be 5 different neurons all looking at the same region in the input volume

[Credit: Karpathy - Standford – CS231 course]

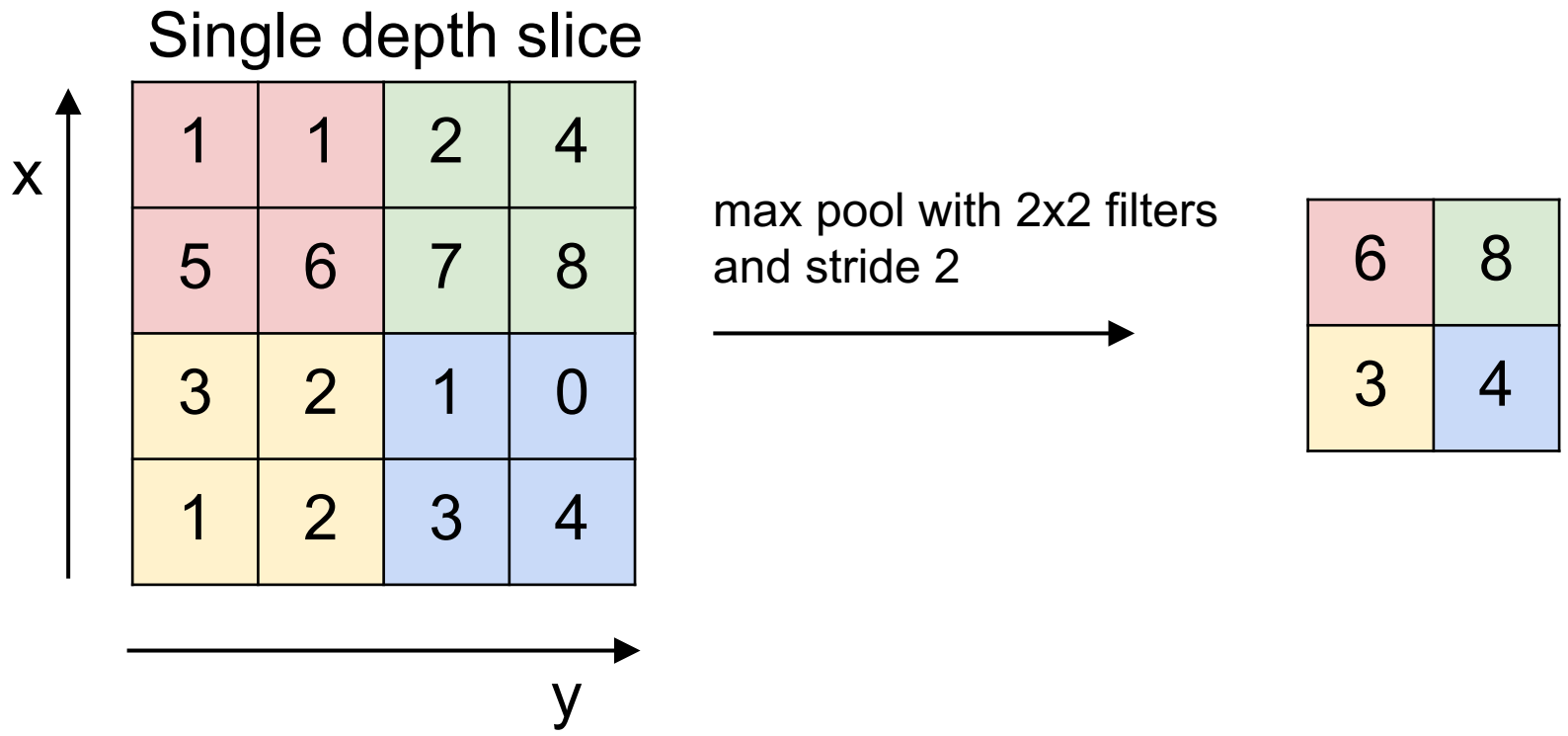
Pooling Layer

- makes the representations smaller and more manageable



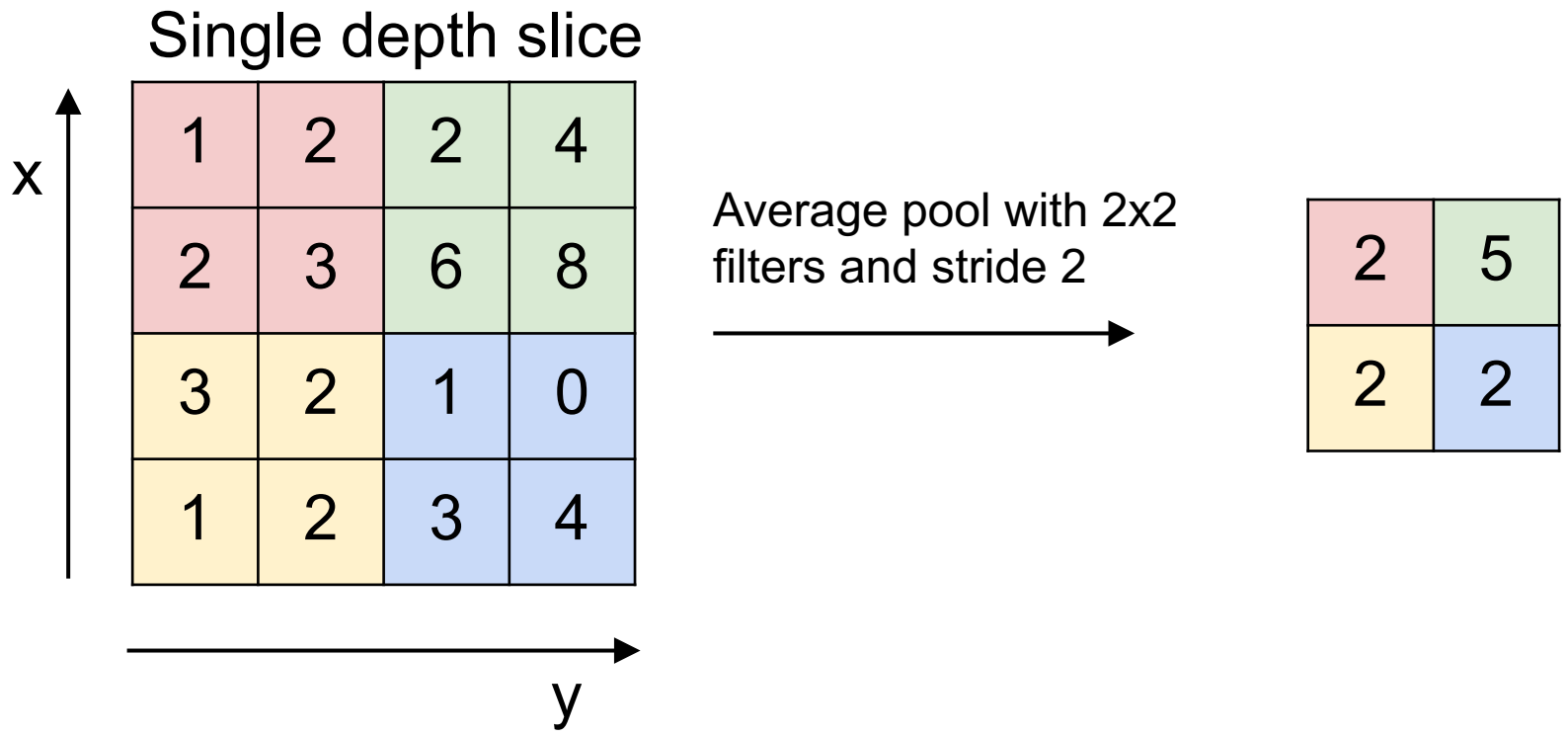
[Credit: Karpathy - Standford – CS231 course]

Max Pooling



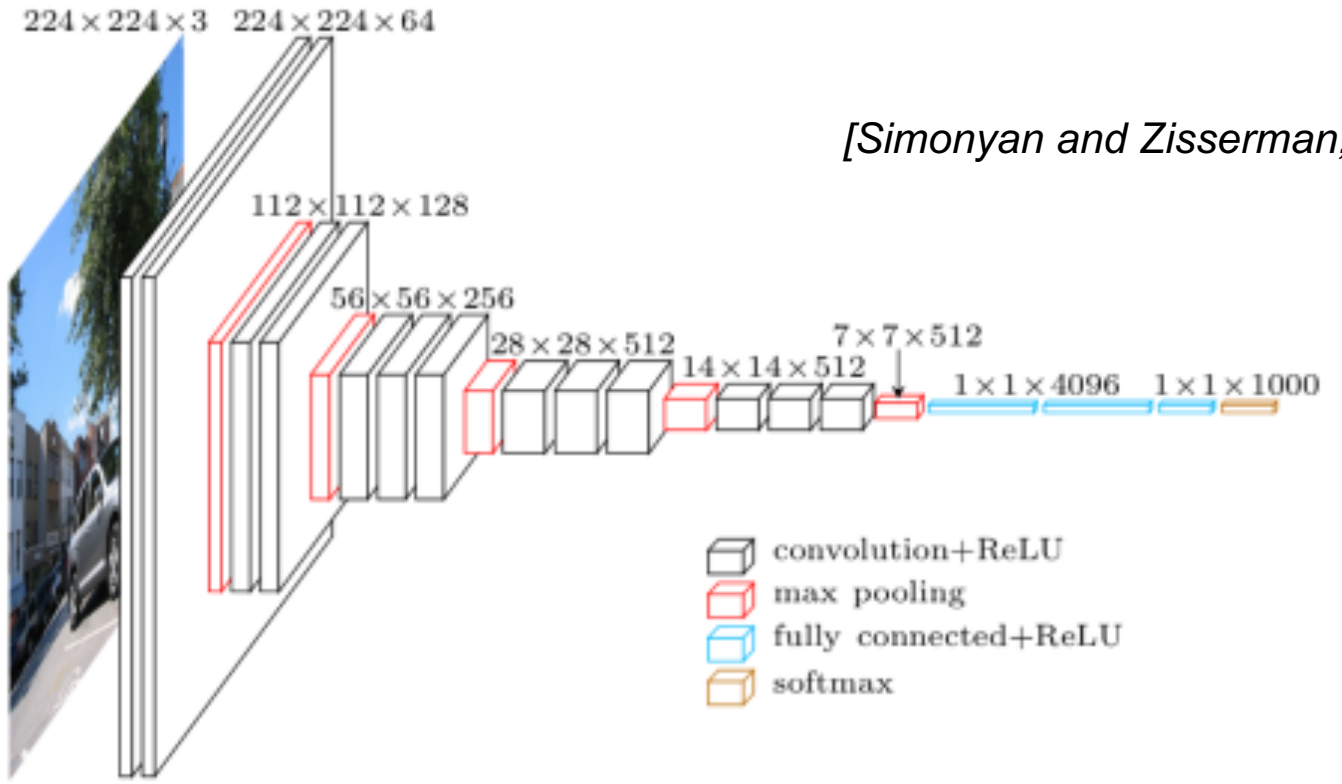
[Credit: Karpathy - Standford – CS231 course]

Average Pooling



Example of the VGG 16 architecture

- Max pooling is used at multiple locations to reduce the dimensions



[Simonyan and Zisserman, 2014]

➢ This requires to resize the image to a fix input size

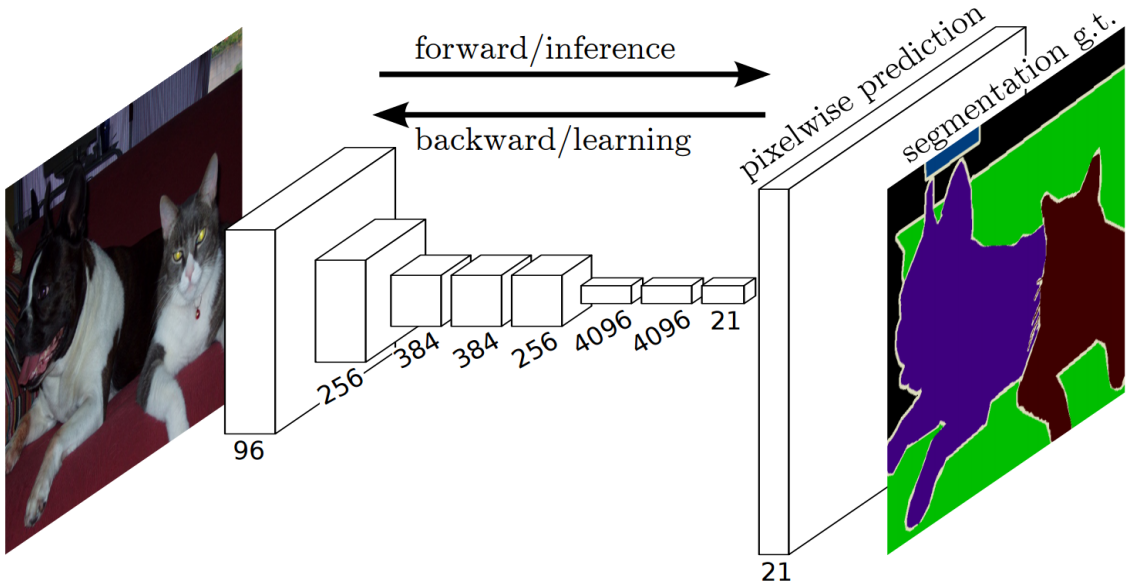
Fully convolutional networks

Definition: A fully convolutional CNN or FCN

- All the learnable layers are convolutional
- i.e. it doesn't have any fully connected layer

Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015

Initially proposed to perform semantic segmentation, but useful elsewhere



➢ Can be applied to images of any size = no image distortion

Maximum activations of convolutions (MAC)

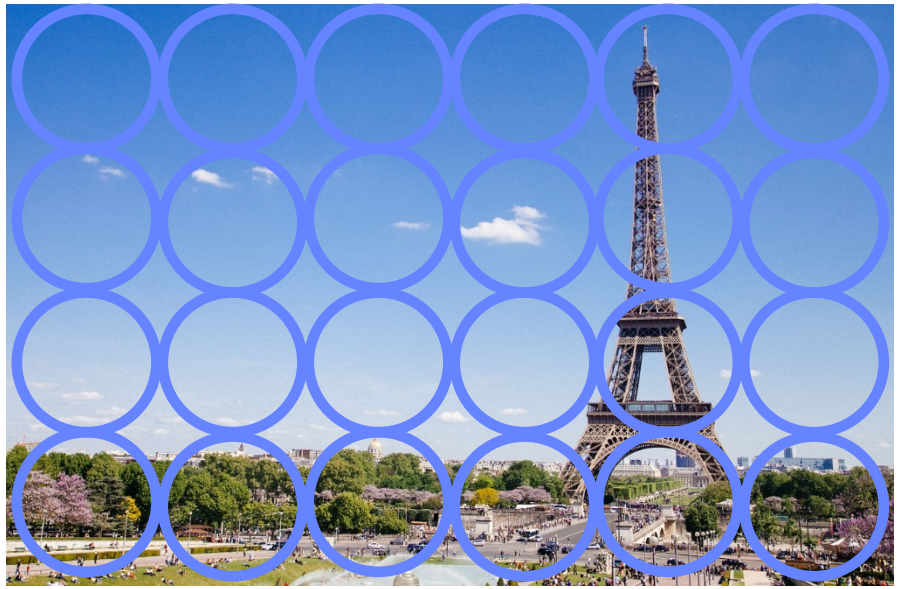
MAC descriptor (1/2)

[Tolias et al, ICLR16]

Input image

CNN

Local features

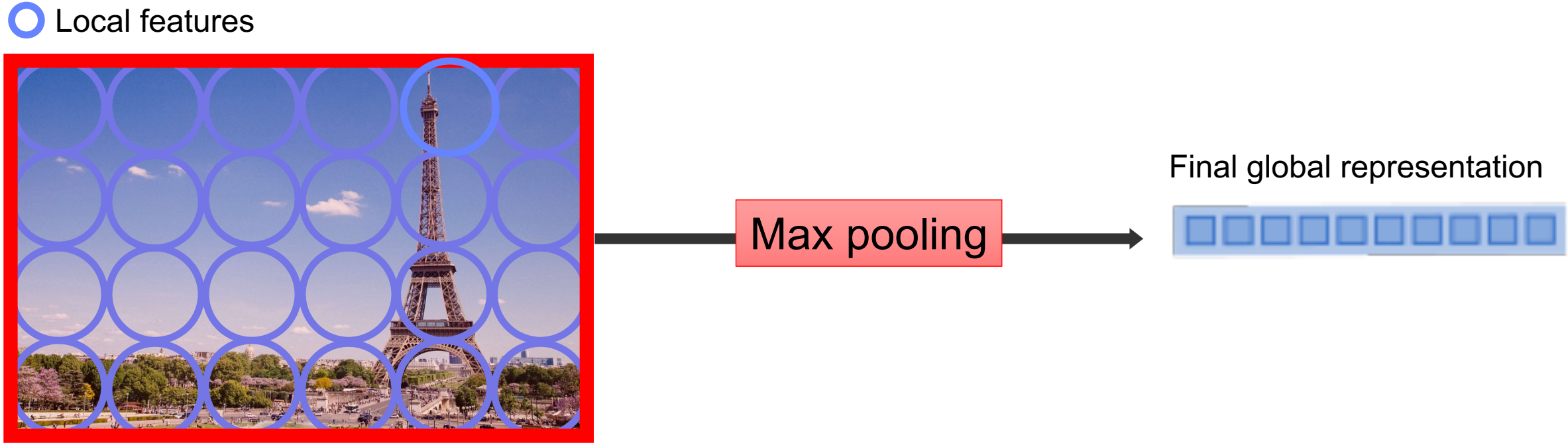


- Can be applied to images of any size = no image distortion

Maximum activations of convolutions (MAC)

MAC descriptor (2/2)

[Tolias et al, ICLR16]



- Can be applied to images of any size = no image distortion

Regional maximum activations of convolutions (R-MAC)

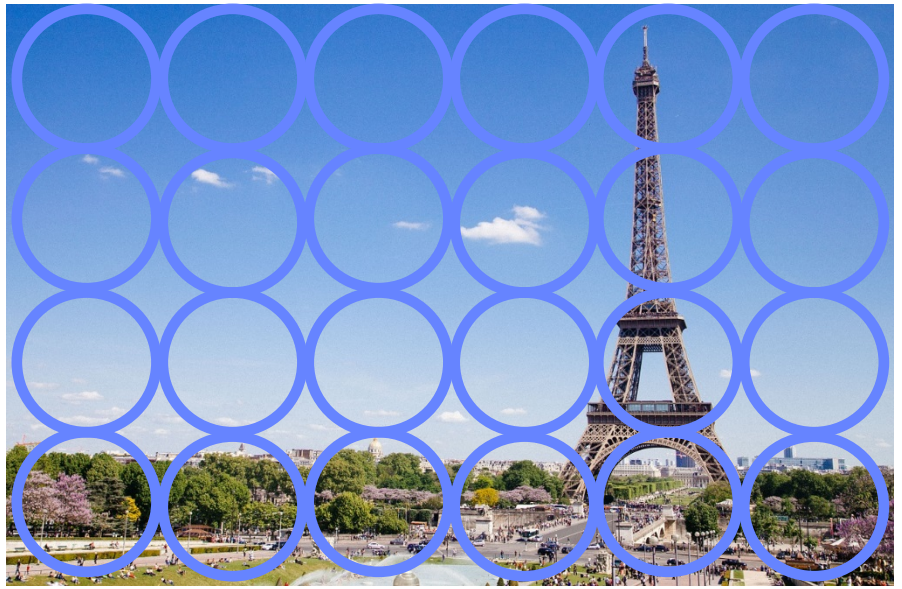
R-MAC descriptor (1/2)

[Tolias et al, ICLR16]

Input image

CNN

Local features



- Can be applied to images of any size = no image distortion

Rappel cours 3 !

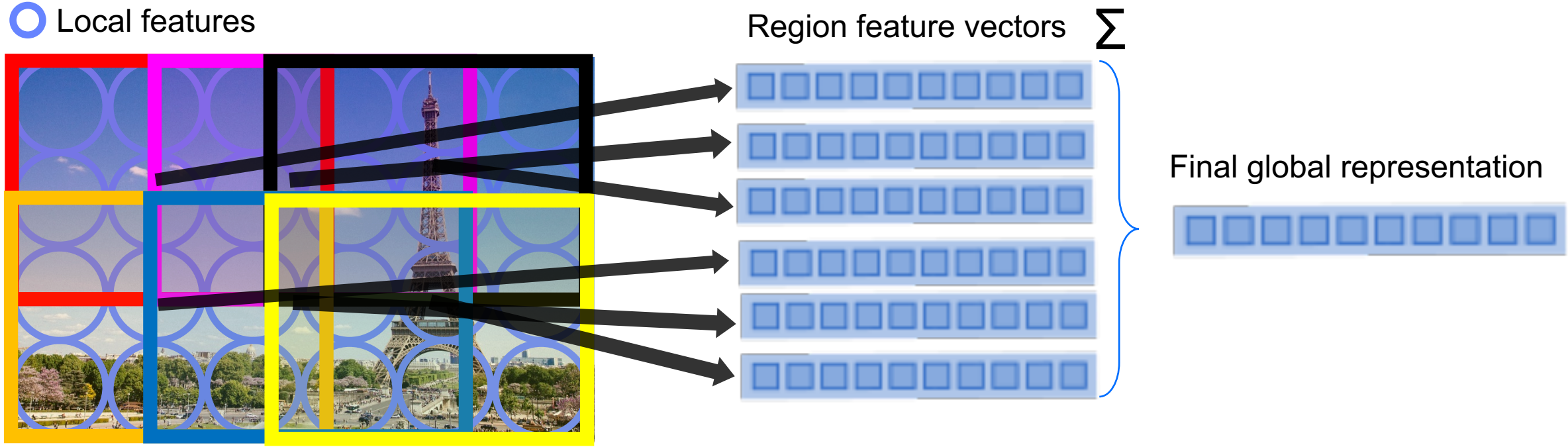
2. Architecture

R-MAC descriptor (2/2)

[Tolias et al, ICLR16]

Advantages

- no aspect ratio distortion
- can encode high resolution images
- fast comparison with the dot product



NAVER LABS EUROPE

Different aggregation techniques

MAC

[Tolias et al. ICLR16]

- Spatial max-pooling

RMAC

- Spatial max-pooling per region + sum-pooling over the regions

SPoC

[Babenko and Lempitsky. ICCV15]

- Sum-pooling aggregation over convolutional features
- Center prior for spatial weighting and uniform channel weighting

CroW

[Kalantidis et al. W@ECCV16]

- Non-parametric scheme for spatial- and channel-wise weighting before sum-pooling aggregation
- Boosts the effect of highly activate spatial responses and regulates burstiness

Different aggregation techniques

Generalized-Mean pooling

[Radenovic et al. PAMI18]

- Trainable Generalized-Mean pooling layer that generalizes max and average pooling

Assuming that the network output consists of K activation maps (i.e. 2D feature maps)

$$\mathbf{f}^{(g)} = [\mathrm{f}_1^{(g)} \dots \mathrm{f}_k^{(g)} \dots \mathrm{f}_K^{(g)}]^\top, \quad \mathrm{f}_k^{(g)} = \left(\frac{1}{|\mathcal{X}_k|} \sum_{x \in \mathcal{X}_k} x^{p_k} \right)^{\frac{1}{p_k}}$$

Notes

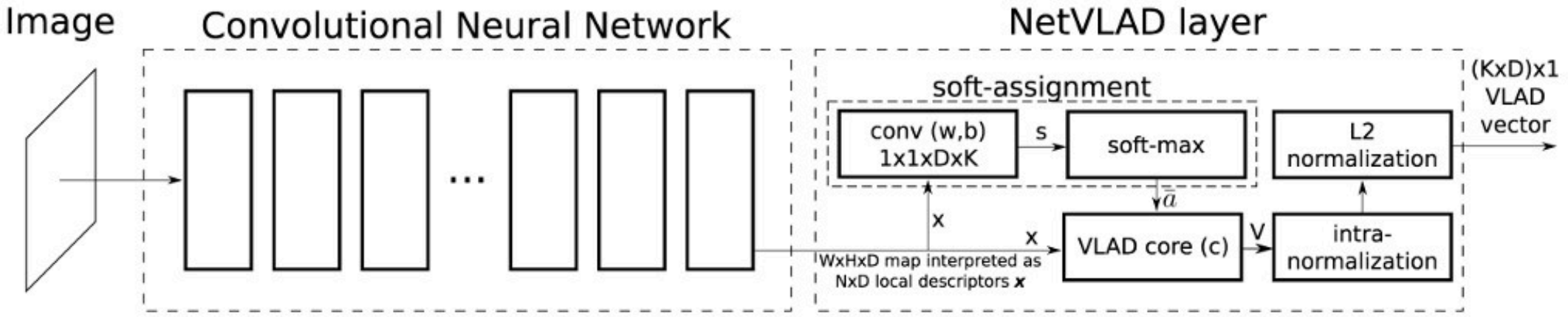
- **Top performing approach**
- When p $\rightarrow \infty$ special case of max pooling
- When p = 1 special case of average pooling

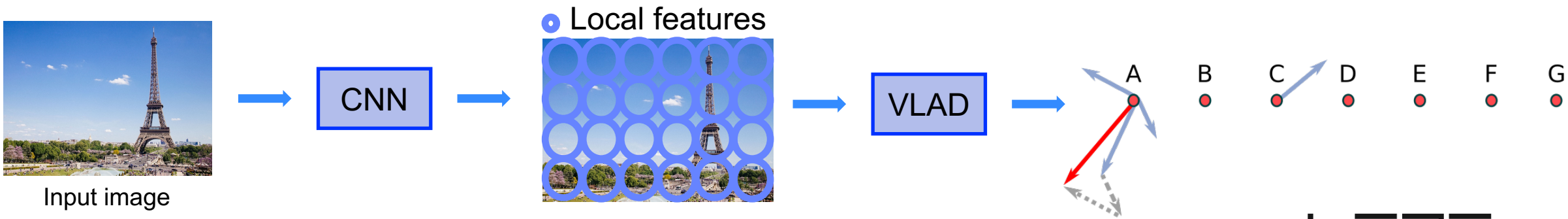
Combining CNN and VLAD: **NetVLAD**

Principle: “VLAD pooling”

- VLAD block at the end of CNN

[Arandjelovic et al. CVPR16]





[Gong et al. ECCV14, Paulin et al ICCV15]

NAVER LABS EUROPE

Merci