COURS RDFIA deep Image
https://cord.isir.upmc.fr/teaching-rdfia/

Matthieu Cord
Sorbonne University
Course Outline

1. **Computer Vision basics**: Visual (local) feature detection and description, Bag of Word Image representation
2. **Introduction to Neural Networks (NNs)**
3. **Machine Learning basics**: Risk, Classification, Datasets, benchmarks and evaluation, Linear classification (SVM)
4. **Neural Nets for Image Classification**
5. **Large scale convolutional neural nets**
6. **Vision Transformers**
7. **Transfer learning and domain adaptation**
8. **Segmentation and Detection**
9. **Generative models with GANs**
10. **Control – AI Challenges**
11. **Explainable AI, Applications**
12/13 **Bayesian deep learning**
14. **Robustness**

https://cord.isir.upmc.fr/teaching-rdfia/
COMPUTER VISION:
(Processing, analyzing and) understanding visual data
=> WHERE ARE WE NOW?

Facts: Exponential increase in quantity of images/videos taken across the world

- YouTube: 500h of video / min
- Facebook: 300M photos / day

Source (many slides): Cornell CV course
Deployed: Optical character recognition (OCR)

- If you have a scanner, it probably came with OCR software

Digit recognition, AT&T labs
http://www.research.att.com/~yann/

License plate readers
http://en.wikipedia.org/wiki/Automatic_number_plate_recognition

Automatic check processing

Source: S. Seitz
Deployed: Face detection

- Cameras now detect faces
  - Canon, Sony, Fuji, ...
Deployed & Significant progress: Face Recognition
Significant progress: Recognizing objects

Mask R-CNN. Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick. ICCV 2017
Ex: Recognition-based product search
Recognition-based product search
Recognition-based product search
Significant progress: Species recognition

iNaturalist dataset

Challenges:
• fine-grained recognition
• Detecting rare concepts
Challenges: Fully autonomous driving
Challenges: Medical Imaging, Health

**Fig. 1: Glioma sub-regions.** Shown are image patches with the tumor sub-regions that are annotated in the different modalities (top left) and the final labels for the whole dataset (right). The image patches show from left to right: the whole tumor (yellow) visible in T2-FLAIR (Fig.A), the tumor core (red) visible in T2 (Fig.B), the enhancing tumor structures (light blue) visible in T1Gd, surrounding the cystic/necrotic components of the core (green) (Fig. C). The segmentations are combined to generate the final labels of the tumor sub-regions (Fig.D): edema (yellow), non-enhancing solid core (red), necrotic/cystic core (green), enhancing core (blue). (Figure taken from the BraTS IEEE TMI paper.)
Challenges: Medical Imaging, Health

Building system to detect Covid in chest x rays
What should a metric measure?
Accuracy = P(pred. label == true label)
Accuracy of candidate system = 95%
Is this good? Did it actually help / work?

Artificial intelligence / Machine learning

Hundreds of AI tools have been built to catch covid. None of them helped.

Some have been used in hospitals, despite not being properly tested. But the pandemic could help make medical AI better.

by Will Douglas Heaven  July 30, 2021
Typical issues that plague deployment

- Images seen during deployment are very different: **domain shift**
- Meaning of classes etc. change: **concept drift**
- Unforeseen circumstances, e.g., new classes: **open world**
Challenges: Integrating Vision and Action, Robotics

Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, Jitendra Malik
CVPR 2017
Challenges: Understanding complex situations / Reasoning
Challenges: Visual Reasoning

VQA task: Why is this funny?

Andrej Karpathy
Challenges: Generative models for images - edition, manipulation
Course Outline

1. Computer Vision:

   Visual (local) feature detection and description,
   Bag of Word Image representation
Local feature detection and description

Points/Regions of Interest detection

One example: Corner detection (Harris corner detector)
Corner detection

- Corner point: singular point highly informative, rare, ...
- Basic idea for Algo: For each pixel \((x,y)\) from image \(I\), *translating* a centered window: Iff \((x,y)\) is a corner, it should cause large differences in patch appearance (whatever the translation)
Corner Detection: Basic Idea

"flat" region: no change in all directions

"edge": no change along the edge direction

"corner": significant change in all directions

Corner detection op == For all pix, shift a window in any direction, keep the ones that give a large change in intensity
Harris corner detection: algo1

Consider a pix \((x,y)\), a small window \(W\), a shifting vector \((u,v)\):

- how do the pixels in \(W\) change?
- compare each pixel before and after by summing up the squared differences (SSD)
- this defines an SSD “error” \(E(u,v)\):

\[
E(u, v) = \sum_{(x,y) \in W} [I(x + u, y + v) - I(x, y)]^2
\]

- To select \((x,y)\) as corner, \(E(u,v)\) has to be as high as possible for all shifting dir \((u,v)\)!
Simplify $E(u,v)$? Small motion assumption

Taylor Series expansion of $I$:

$$I(x+u, y+v) = I(x, y) + \frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \text{higher order terms}$$

If the motion $(u,v)$ is small, then first order approximation is good

$$I(x + u, y + v) \approx I(x, y) + \frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v$$

$$\approx I(x, y) + [I_x \ I_y] \begin{bmatrix} u \\ v \end{bmatrix}$$

shorthand: $I_x = \frac{\partial I}{\partial x}$

Plugging this into the formula on the previous slide...
Simplify $E(u,v)$? Small motion assumption

\[
E(u, v) = \sum_{(x,y) \in W} [I(x + u, y + v) - I(x, y)]^2
\]

\[
\approx \sum_{(x,y) \in W} [I(x, y) + I_x u + I_y v - I(x, y)]^2
\]

\[
E(u, v) \approx \sum_{(x,y) \in W} [I_x u + I_y v]^2
\]

\[
\approx Au^2 + 2Bu v + Cv^2
\]

\[
A = \sum_{(x,y) \in W} I_x^2,
\quad B = \sum_{(x,y) \in W} I_x I_y,
\quad C = \sum_{(x,y) \in W} I_y^2
\]

$E(u,v)$ is locally approximated as a quadratic error function $\Rightarrow$ Once $A$, $B$ and $C$ computed, very fast to compute $E(u,v)$ for many $(u,v)$!

ALGO 2!
Interpreting the second moment matrix

\[
M = \sum_{x,y} \begin{bmatrix}
I_x^2 & I_x I_y \\
I_x I_y & I_y^2
\end{bmatrix} = \begin{bmatrix}
A & B \\
B & C
\end{bmatrix}
\]

\[
E(u,v) \approx [u \ v] M [u \\
v]
\]

Recall that we want \(E(u,v)\) to be as large as possible for all \(u,v\)

What does this mean in terms of \(M\)?
\[ E(u, v) \approx \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} A & B \\ B & C \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \]

\[ A = \sum_{(x,y) \in W} I_x^2 \]
\[ B = \sum_{(x,y) \in W} I_x I_y \]
\[ C = \sum_{(x,y) \in W} I_y^2 \]

Flat patch:

\[ I_x = 0 \]
\[ I_y = 0 \]
\[ E(u, v) \approx \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} A & B \\ B & C \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \]

\[
A = \sum_{(x,y) \in W} I_x^2 \\
B = \sum_{(x,y) \in W} I_x I_y \\
C = \sum_{(x,y) \in W} I_y^2
\]

Vertical edge: \( I_y = 0 \)

\[
M = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix}
\]

\[
M \begin{bmatrix} 0 \\ v \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\]

\[ E(0, v) = 0 \quad \forall v \]
\[ E(u, v) \approx \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} A & B \\ B & C \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \]

\[ A = \sum_{(x, y)\in W} I_x^2 \]
\[ B = \sum_{(x, y)\in W} I_x I_y \]
\[ C = \sum_{(x, y)\in W} I_y^2 \]

Horizontal edge: \( I_x = 0 \)

\[ M = \begin{bmatrix} 0 & 0 \\ 0 & C \end{bmatrix} \]
\[ M \begin{bmatrix} u \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \]
\[ E(u, 0) = 0 \quad \forall u \]
What about edges in arbitrary orientation?
\[ E(u, v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} \]

\[ M \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow E(u, v) = 0 \]

\[ M \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \iff E(u, v) = 0 \]

Solutions to \( Mx = 0 \) are directions for which \( E \) is 0: window can slide in this direction without changing appearance.
$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$

Solutions to $Mx = 0$ are directions for which $E$ is 0: window can slide in this direction without changing appearance.

For corners, no such directions exist.
Eigenvalues and eigenvectors of $M$

- $Mx = 0 \Rightarrow Mx = \lambda x$: $x$ is an eigenvector of $M$ with eigenvalue 0
- $M$ is 2 x 2, so it has 2 eigenvalues $(\lambda_{\text{max}}, \lambda_{\text{min}})$ with eigenvectors $(x_{\text{max}}, x_{\text{min}})$
- $E(x_{\text{max}}) = x_{\text{max}}^T M x_{\text{max}} = \lambda_{\text{max}} \| x_{\text{max}} \|^2 = \lambda_{\text{max}}$
  (eigenvectors have unit norm)
- $E(x_{\text{min}}) = x_{\text{min}}^T M x_{\text{min}} = \lambda_{\text{min}} \| x_{\text{min}} \|^2 = \lambda_{\text{min}}$
Eigenvalues and eigenvectors of $M$

$$E(u, v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$$

- Define shift directions with the smallest and largest change in error
- $x_{\text{max}}$ = direction of largest increase in $E$
- $\lambda_{\text{max}}$ = amount of increase in direction $x_{\text{max}}$
- $x_{\text{min}}$ = direction of smallest increase in $E$
- $\lambda_{\text{min}}$ = amount of increase in direction $x_{\text{min}}$
Interpreting the eigenvalues

\[ \lambda_{\text{max}}, \lambda_{\text{min}} \text{ are small; } E \text{ is almost } 0 \text{ in all directions} \]

\[ \lambda_{\text{max}} \approx \lambda_{\text{min}} \gg 0 \]

E very high in all directions

\[ \lambda_{\text{max}} \gg \lambda_{\text{min}}, \lambda_{\text{min}} \approx 0 \]

E remains close to 0 along \( x_{\text{min}} \)
Corner detection: M-based algo

How are $\lambda_{\text{max}}$, $x_{\text{max}}$, $\lambda_{\text{min}}$, and $x_{\text{min}}$ relevant for feature detection?

- Need a feature scoring function

Want $E(u, v)$ to be large for small shifts in all directions

- the minimum of $E(u, v)$ should be large, over all unit vectors $[u \ v]$

- this minimum is given by the smaller eigenvalue ($\lambda_{\text{min}}$) of $M$

Good detector: $\lambda_{\text{min}} > \text{threshold}$
Corner detection summary

Algo3 (M-based)

• Compute the gradient at each point in the image
• Create the $M$ matrix from the entries in the gradient
• Compute the eigenvalues of $M$
• Find points with large response ($\lambda_{\text{min}} > \text{threshold}$)
• Choose those points where $\lambda_{\text{min}}$ is a local maximum as features
Corner detection summary

Algo3 (M-based)
- Compute the gradient at each point in the image
- Create the $H$ matrix from the entries in the gradient
- Compute the eigenvalues
- Find points with large response ($\lambda_{\text{min}} > \text{threshold}$)
- Choose those points where $\lambda_{\text{min}}$ is a local maximum as features
Algo4: The Harris operator

Algo3 still expensive because explicit eigen-decomposition of $M$.

“Harris operator” for corner detection is a variant of the Algo3 ($\lambda_{\text{min}}$ based)

Heuristic criterion $R$ using matrix properties but not explicit decomposition:

$$R = \det(M) - \alpha \, \text{trace}(M)^2 = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

- The $\text{trace}$ is the sum of the diagonals, i.e., $\text{trace}(H) = h_{11} + h_{22}$
- Very similar to $\lambda_{\text{min}}$ but less expensive (no square root)
- Called the “Harris Corner Detector” or “Harris Operator”
- Simple threshold $R>0$:
The Harris operator

Harris operator

$\lambda_{\text{min}}$
Harris detector example
f value (red high, blue low)
Threshold ($f > value$)
Find local maxima of $f$
Harris features (in red)
Local feature detection

Looking for repeatability
Local feature detection

One example: Corner detection (Harris corner detector)
Many other Points/Regions of Interest detectors

Sparse, at interest points
Dense, uniformly
Randomly
Course Outline

1. Computer Vision Introduction:
   - Visual (local) feature detection,
   - Visual (local) feature description,
   - Bag of Word Image representation
Local feature description

Many Points/Regions of Interest descriptors

One example: SIFT descriptor

Local description (always looking for invariance)

SIFT descriptors/features
Feature descriptors

• Expected properties?
  – Similar patches => close descriptors
  – Invariance (robustness) to geom. transformation: rotation, scale, view point, luminance, semantics? …
BoF: (First) Image representation

Sparse, at interest points

Dense, uniformly

Randomly

Multiple interest operators

Feature extraction

A bag of features BoF

© F-F. Li, E. Nowak, J. Sivic
BoF -- Image representation

- Image similarity based on matching of local features + voting
Applications to Image Retrieval

Query

Target (if in)
Most similar to Q
+ infos: The Wedding at Cana -- Véronèse
Image Retrieval

- Context: Instance search (second example)
Advanced Visual Understanding

Two pizzas sitting on top of a stove top oven
Image Understanding

• Focus of this course: recognition, classification and understanding
• Fundamental Pbs:
  – Image representation,
  – Data similarity,
  – Decision function
• Examples of applications:
  – (Health) Medical imagery
  – (Mobility) Autonomous driving/Robotics
  – (Security/entertainment) Face/ Human action recognition
  – (Physics, Astronomy, Bio ... ) Pattern recognition
Course Outline

1. Computer Vision Introduction:
   Visual (local) feature detection and description,
   Bag of Word Image representation

   1. Introduction to Bag of Words
   2. Visual Dictionary
   3. Image signature
   4. Whole recognition pipeline
Bag of Feature (BoF) Model
Bag of Words representation

- BoF
  - Local signatures: not a scalable representation
  - Not a *semantic* representation

- Model to represent images for categorization: « Bag of Words BoW »
- BoW model computed from BoF (Bag of features)
Bag of Words (BoW) model: basic explication with textual representation and color indexing

Of all the sensory impressions proceeding to the brain, the visual experiences are the dominant ones. Our perception of the world around us is based essentially on the messages that reach the brain from our eyes. For a long time it was thought that nerve, image Hubel, Wiesel

China is forecasting a trade surplus of $90bn (£51bn) to $100bn this year, a threefold increase on 2004's $32bn. The Commerce Ministry said the surplus would be created by a predicted 30% jump to rise further in value.

Comparing 2 docs using visual/color/word occurrences

Slide credit L. Fei-Fei
Questions:

1. Which dictionary?
2. How to project the BoF onto the dico
3. How to compute the histogram?
Course Outline

1. Computer Vision Introduction:
   Visual (local) feature detection and description,
   Bag of Word Image representation

   1. Introduction to Bag of Words
   2. Visual Dictionary
   3. Image signature
   4. Whole recognition pipeline
Visual space clustering

1. Extraction of local features (pattern/visual words) in images
   • Training dataset in classification
   • Image dataset in retrieval

2. Clustering of feature space

Training set but no labels => UNSUPERVISED Learning
Visual space clustering

• Many algorithms for clustering:
  • K-Means
  • Vectorial Quantization
  • Gaussian Mixture Models
  • ...
Clustering with K clusters

Input: set of n points \( \{x_j\}_n \) in \( \mathbb{R}^d \)

Goal: find a set of K (K<<n) points \( w_1, \ldots, w_K \)

that gives an approximation of the n input points,

ie. minimizing mean square error \( C(w) \):

\[
C(w) = \sum_{i=1}^{n} \min_k \| x_i - w_k \|^2
\]

At k fixed, complexity is \( O(n^{(Kd+1)\log(n)}) \)

A lot of strategies to approximate the global optimization problem
Clustering with K clusters

\[ C'(w) = \sum_{i=1}^{n} \min_k ||x_i - w_k||^2 \]

**K-means Algorithm:**

1. (Re)assign each point \( x_i \) to the cluster \( s_i \) with the center \( w_{s_i} \) so that \( \text{dist}(x_i, w_{s_i}) \) is less than \( \text{dist} \) from \( x_i \) to any other clusters.

2. Move all \( w_k \) inside each cluster as the new barycenter from all the points assigned to the cluster \( k \) (equ. to minimize the corresponding mean square error).

3. Go to step 1 if some points changed clusters during the last iteration.

Output: the set of the final K cluster centers \( \{c_k = w_k\} \)
K-means: why it is successful?

Consider an arbitrary cluster assignment \( s_i \).

\[
C(w) = \sum_{i=1}^{n} \min_k \|x_i - w_k\|^2 = \sum_{i=1}^{n} \|x_i - w_{s_i}\|^2 - \sum_{i=1}^{n} \|x_i - w_{s_i}\|^2 - \min_k \|x_i - w_k\|^2
\]

\( \mathcal{L}(s,w) \)

\( \mathcal{D}(s,w) \geq 0 \)

1. Change \( s_i \) to minimize \( \mathcal{D} \) leaving \( C(w) \) unchanged.

2. Change \( w_k \) to minimize \( \mathcal{L} \). Meanwhile \( \mathcal{D} \) can only increase.

\( \mathcal{L} \)

\( \mathcal{D} = 0 \)

\( \mathcal{D} \geq 0 \)

© L. Botou
Clustering

- **K-means** :
  - **Pros**
    - Simplicity
    - Convergence (local min)
  - **Cons**
    - Memory-intensive
    - Depending on K
    - Sensitive to initialization
    - Sensitive to artifacts
    - Limited to spherical clusters
    - Concentration of clusters to areas with high densities of points
      (Alternatives: radial based methods)
  - **K-Means deeply used in practice**
Clustering

- Uniform / K-means / radius-based:
  - Radius-based clustering assigns all features within a fixed radius of similarity $r$ to one cluster.
Dictionary = K Visual words

Extraction

Clustering

Centers = dico. Visual words

Dico examples

Dico extraction
Course Outline

1. Computer Vision Introduction:
   Visual (local) feature detection and description,
   Bag of Word Image representation

   1. Introduction to Bag of Words
   2. Visual Dictionary
   3. Image signature
   4. Whole recognition pipeline
Bag-of-Words (BoW) image signature

- For each image:
  - For each local feature: find the closest visual word
  - Increase the corresponding bin in histogram of visual dico

- Image signature (global Index):
  - Vector (histogram of M bins)
  - M= dimension K = dico size
  - Each term represents a Likelihood to get this visual word
Bag-of-Words (BoW) image signature

- Original BoW strategy: **hard assignment/coding**
  - Find the closest cluster for each feature
  - Assign a fix weight (e.g. 1)
Bag-of-Words (BoW) image signature

**Sum pooling**: initial BoW strategy (just counting occurrences of words in the document)

Classical BoW = hard coding + sum pooling

1. Find the closest cluster for each feature
2. Assign a fix weight (*e.g.* 1) to this cluster
BoW: the math

Image features: \( X = \{ x_j \in \mathbb{R}^d \}, j \in \{1; N\} \)

Centers: \( C = \{ C_m \}, m \in \{1; M\} \)

Coding:
\[
f : \mathbb{R}^d \rightarrow \mathbb{R}^M
\]
\[
x_j \rightarrow f(x_j) = \alpha_j = \{ \alpha_{m,j} \}, \quad m \in \{1; M\}
\]

Hard coding: \( f = f_Q \) assigns a constant weight to its closest center:
\[
f_Q(x_j)[m] = \begin{cases} 1 & \text{if } m = \arg\min_{k} \| x_j - c_k \|^2 \\
0 & \text{otherwise} \end{cases} \quad k \in \{1; M\}
\]
BoW: the math

$$\mathbf{H} = \begin{bmatrix}
\alpha_{1,1} & \cdots & \alpha_{1,j} & \cdots & \alpha_{1,N} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
\alpha_{m,1} & \cdots & \alpha_{m,j} & \cdots & \alpha_{m,N} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
\alpha_{M,1} & \cdots & \alpha_{M,j} & \cdots & \alpha_{M,N}
\end{bmatrix}
\Rightarrow g: pooling$$

$$f: coding$$
BoW: the math

- Global Index: image likelihood to get each visual word
- Several strategies to aggregate the projections: pooling

\[ g : \mathbb{R}^N \rightarrow \mathbb{R} \]
\[ \alpha_m = \{\alpha_{m,j}\}, j \in \{1; N\} \rightarrow g(\alpha_m) = z_m \]
**BoW: the math**

\[
\begin{bmatrix}
\alpha_{1,1} & \cdots & \alpha_{1,j} & \cdots & \alpha_{1,N} \\
\vdots & & \vdots & & \vdots \\
\alpha_{m,1} & \cdots & \alpha_{m,j} & \cdots & \alpha_{m,N}
\end{bmatrix}
\]

\[
H = \begin{bmatrix}
\alpha_{1,1} & \cdots & \alpha_{1,j} & \cdots & \alpha_{1,N} \\
\vdots & & \vdots & & \vdots \\
\alpha_{m,1} & \cdots & \alpha_{m,j} & \cdots & \alpha_{m,N}
\end{bmatrix}
\Rightarrow g: pooling
\]

\[
f: coding
\]
BoW: the math

BoW Sum pooling:

\[ z_m = g(\alpha_m) = \sum_{j=1}^{N} \alpha_{m,j} = \sum_{j=1}^{N} f_Q(x_j)[m] \]

\[ z_m = \sum_{j=1}^{N} \begin{cases} 
1 & \text{if } m = \arg\min_{k \in \{1; M\}} \|x_j - c_k\|^2 \\
0 & \text{otherwise}
\end{cases} \]
BoW: the math

Work on local descriptors

\[ x_1 \quad x_j \quad x_N \]

\[ \mathbf{H} = \begin{bmatrix}
\alpha_{1,1} & \cdots & \alpha_{1,j} & \cdots & \alpha_{1,N} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
\alpha_{M,1} & \cdots & \alpha_{M,j} & \cdots & \alpha_{M,N}
\end{bmatrix} \]

\[ f: \text{coding} \quad \Downarrow \quad g: \text{pooling} \]

Work on dico

Work on pooling

Work on coding
Bag of Word Image representation

1. Introduction to Bag of Words
2. Dictionary computation
3. Coding of local descriptors
4. Image signature computation: pooling
5. Whole recognition pipeline
Representation

1. feature detection & representation

2. codewords dictionary

3. image representation
Representation

1. feature detection & representation

2. codewords dictionary

Learning and Recognition

3. category models (and/or) classifiers

4. category decision
Generative method:
- graphical models

Discriminative methods:
- SVM, NNs
Learn a classification model to determine the decision boundary

Image classification based on BoW

Training set

BoW histogram vector space

Decision boundary

bird
dog