Modélisation et Reconnaissance des formes
Représenter et comparer des formes

Marie-Odile Berger
http://members.loria.fr/moberger

September 30, 2025

Part I

Quelques considérations générales

Propriétés attendues de la représentation d'une forme

- la taille: Les données sont rarement de taille raisonnable (spectrogramme, images, sequence,...) $\rightarrow$ il faut adopter une représentation des données de taille raisonnable avec le moins possible de perte d'information
- l'invariance: les données peuvent être enregistrées dans des repères différents (ex orientation différente) ou des conditions différentes (météo, jour/nuit, présence humains/voitures...)
  - Utiliser des mesures invariantes (ou le plus possible) pour caractériser des formes.
  - dans les cas complexes, il n'y a pas de caractérisations évidentes de l'invariance. Celles ci **doivent être apprises**.

Type d'invariance visé

Il peut y avoir de simples mouvements de l'objet, des changements de points de vue, des changements d'illumination, des occultations....

Problème (**malédiction (*)**) de la dimension

*: terme inventé par Richard Bellman pour parle de la difficulté de travailler avec des données appartenant à des espaces de grande dimension.

- Représenter une forme par un vecteur de caractéristiques de **petite taille** permet de limiter la complexité des processus
- Un grand vecteur de caractéristiques peut avoir tendance à **modéliser l'accessoire** (le bruit) plutôt que l'essentiel des données.
- **Malédiction**: il faut énormément de données pour obtenir une bonne estimation. Soient 100 observations d'un phénomène faites dans l'intervalle $[0, 1]$. Pour réaliser dans $[0, 1]^{10}$ une couverture équivalente à celle des 100 points il faudrait $100^{10} = 10^{20}$ observations, ce qui est la plupart du temps inenvisageable.

Représenter les données

Objectifs:

- représenter les formes de manière **compacte et discriminante**
- avoir des **métriques** pour évaluer leur similarité

Deux grandes classes de représentation

- Représentations explicites (handcrafted) créées à la main en fonction des connaissances sur le domaine
- Représentations et métriques de comparaison issues des **techniques d'apprentissage**

Difficulté: obtenir l'**invariance souhaitée** par l'application; Exemples: reconnaissance indépendament du point de vue, localisation par l'image indépendante des conditions environnementales (météo, saison...)

Part II

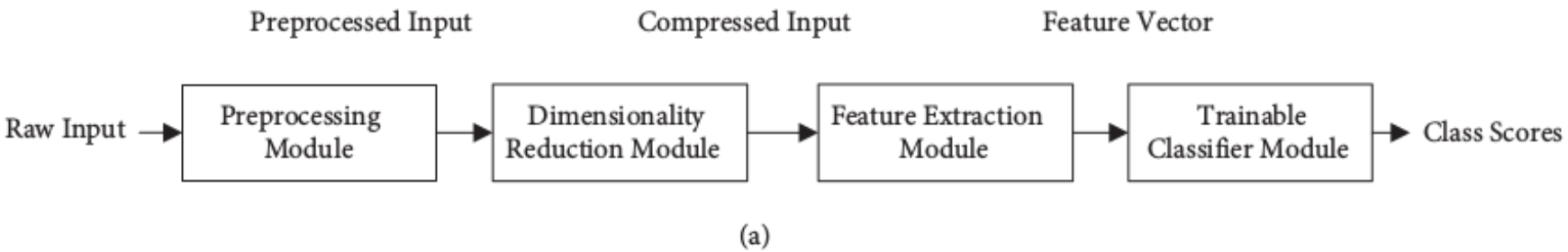
Les approches classiques (Handcrafted) de représentation des formes

Les approches classiques

- classique, conventionnelle, explicite, historique... c'est-à-dire les méthodes existant avant le boom de l'IA
- la représentation des formes est naturelle ou vient de connaissances a priori sur les formes considérées
- exemples:
  - vecteur de paramètres,
  - courbe, courbe paramétrée
  - histogramme
  - ensemble d'indices spécifiques (empreintes digitales, visages,...) fournis par les experts du domaine

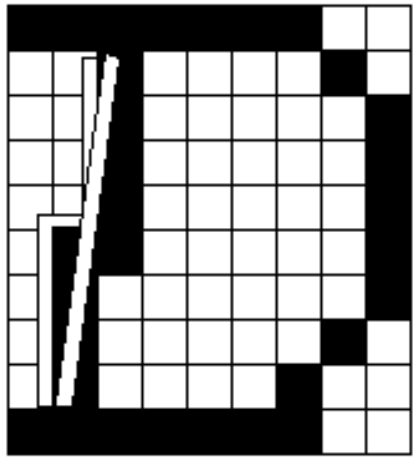
Un système classique de reconnaissance des formes

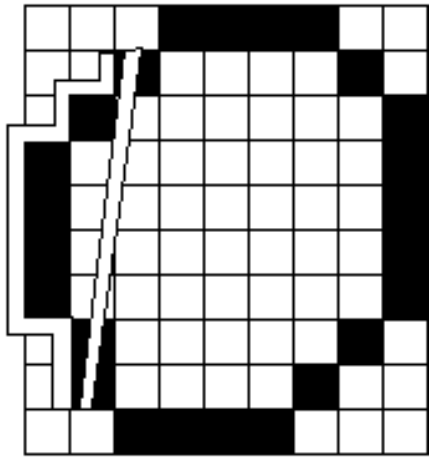
L'approche conventionnelle (ici en vue de classification):



- Les caractéristiques des données sont extraites (de manière statique) **indépendamment** du processus final de classification

Exemples de représentations: les caractères

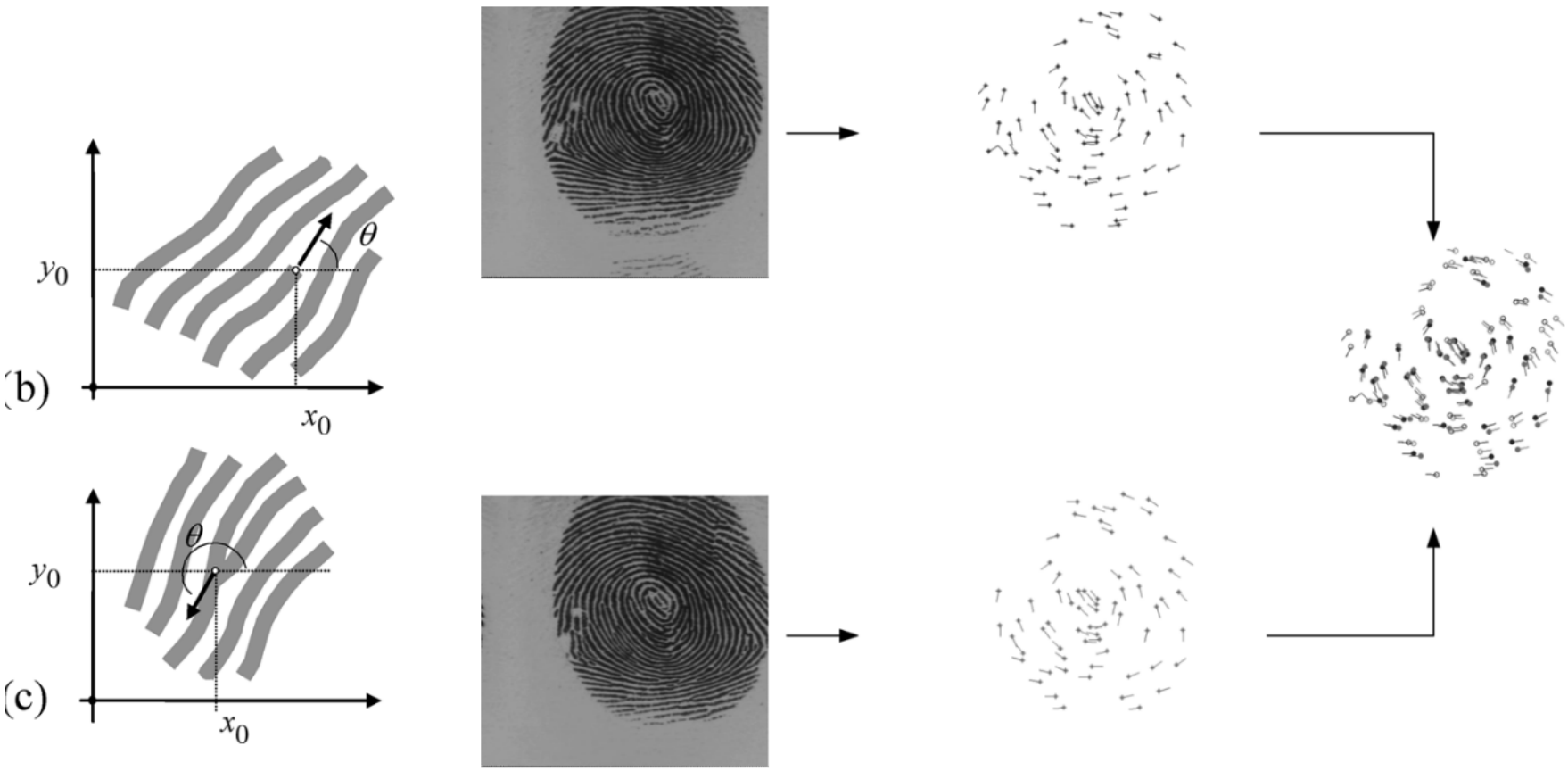




caractéristiques possibles: aire, périmètre, compacité, histogramme ...

Exemples de caractéristiques expert

reconnaissance d'empreintes: mise en correspondance basées sur les *minutia* comme les birfurcations et les points terminaux [7]



les correspondances entre les deux empreintes sont en cercles pleins

Quelle distance pour comparer les formes?

Si la forme est un ensemble de points, sans structure spécifique:

- pour les représentations par un vecteur dans $\mathbb{R}^k$, on utilise les normes classiques, par exemple
  - norme L_2,
  $$||x||_2 = \sqrt{\sum_1^k x_i^2}$$
  - norme L_1,
  $$||x||_1 = \sum |x_i|$$

L_1 est mieux adaptée en cas de données aberrantes mais elle n'est pas dérivable (voir cours estimation robuste).

En pratique

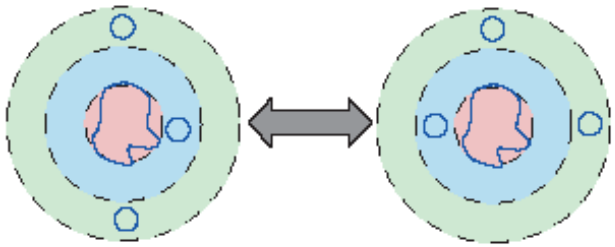
- on utilise des métriques **plus spécifiques** aux formes considérées
- de nombreuses métriques peuvent être définies (qui valent 0 quand les formes sont identiques), mais dont le **comportement (robustesse, invariance, rapidité de calcul) n'est pas le même!**

Distance et similarité

- Mathématiquement, une **distance** est une fonction à valeur positive qui vérifie les propriétés
  1. symmétrie: $d(x, y) = d(y, x)$
  2. $d(x, y) = 0 \rightarrow x = y$
  3. l'inégalité triangulaire : $d(x, y) < d(x, z) + d(z, y)$
- Néanmoins, ce qu'on a tendance à appeler distance en RF vérifie rarement la propriété (2) et encore moins la (3).
- Une mesure de similarité est d'autant plus élevée que les formes se ressemblent. Donc moralement, similarité= 1-distance (si la distance est normalisée).
  - Exemple: le cosinus entre deux vecteurs est une mesure de similarité (2 vecteurs similaires ont un cosinus de 1).

Non respect de (2) $\rightarrow$ création d'une Short list

- Le non respect de la propriété $d(x, y) = 0 \Rightarrow x = y$ est fréquent, en particulier pour les métriques basées sur des histogrammes:
  - exemple 1: deux zones d'une image peuvent avoir le même histogramme sans être identiques
  - exemple 2: métrique=histogramme des angles du contour par rapport au rayon issu du centre de gravité et passant par ce point. Deux formes différentes peuvent partager la même signature

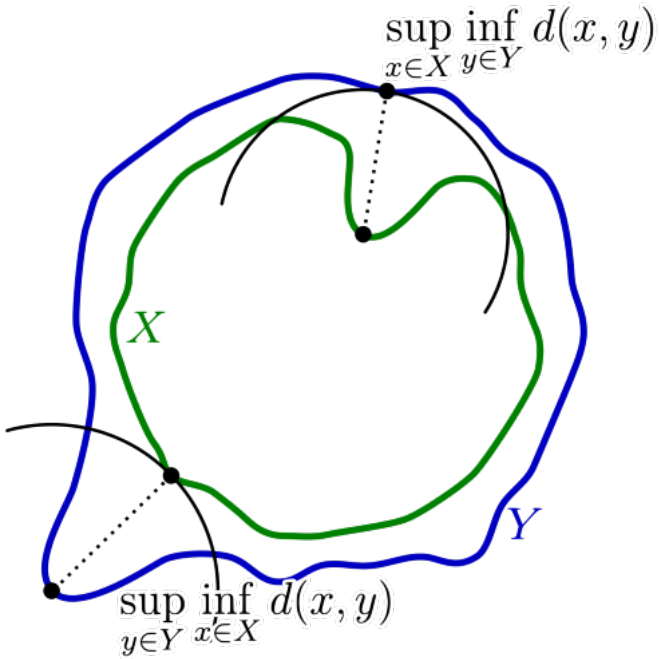


conclusion

Ces métriques servent souvent à créer des "short lists" de manière **rapide**, c-a-d une liste de candidats ressemblant à une requête. Des méthodes **plus fines mais aussi plus lentes**, sont ensuite utilisées pour extraire le meilleur candidat. Voir par exemple l'article *Shape Context* [10]

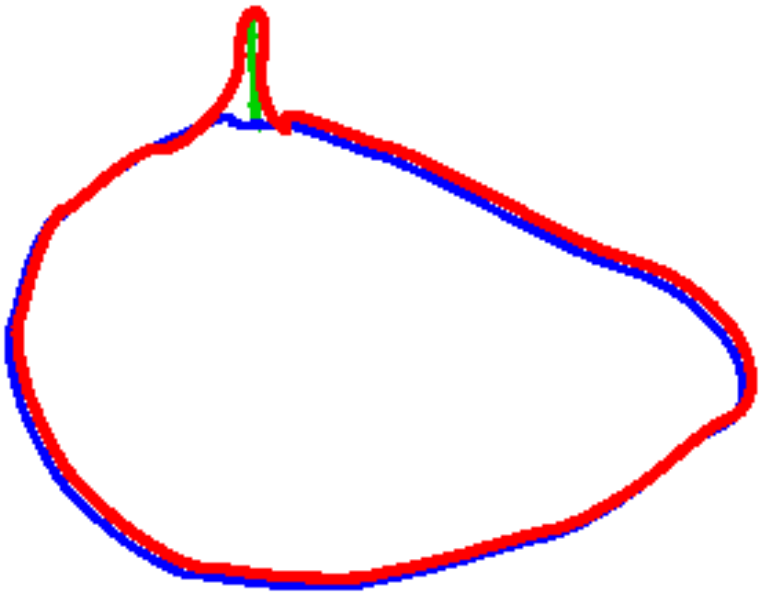
Comparaison ensembliste entre formes: la distance de Hausdorff

- La distance de Hausdorff (qui ne vérifie pas l'inégalité triangulaire)
  - distance entre un point et un ensemble: $d(x, Y) = min_{y \in Y} d(x, y)$
  - distance entre deux ensembles: $d_H(X, Y) = max_{x \in X} d(x, Y)$...
    mais ce n'est pas symmétrique...
  - **distance de Hausdorff** $d_H(X, Y) = max\{max_{x \in X} d(x, Y), max_{y \in Y} d(y, X)\}$



Robustesse de Haussdorf?

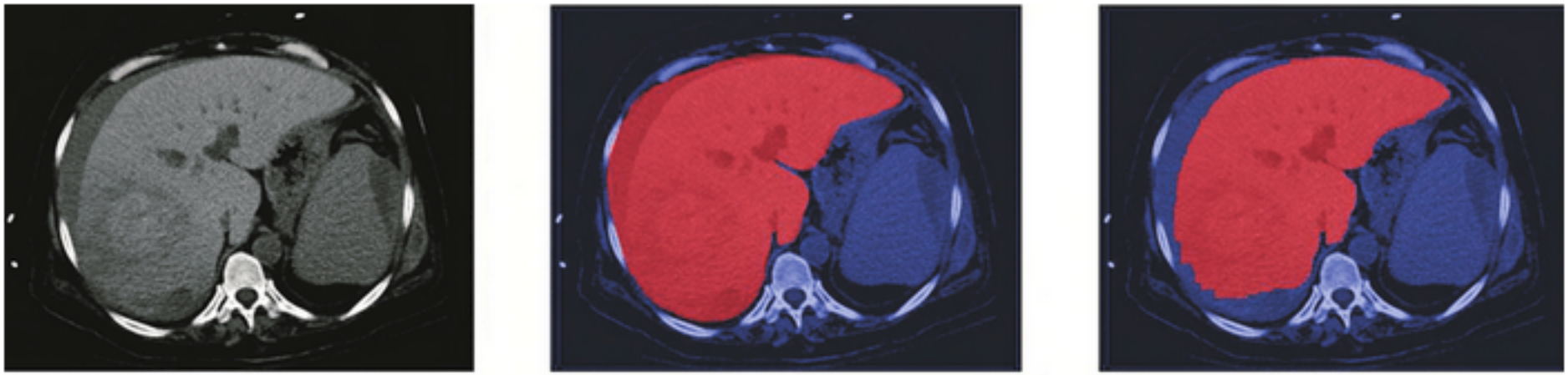
La mesure de Hausdorff n'est pas robuste à de petites variation locales de formes



- différence seulement locale qui impacte la distance de Hausdorff
- divers moyens d'amélioration: $d(X,Y) = \frac{1}{|X|} \sum_{x \in X} min_{y \in Y}\{d(x,y\}$ ou utiliser des méthodes robustes

Et si on compare des objets complexes, structurés?

Exemple: valider une segmentation. Il faut comparer la forme obtenue à une vérité terrain



(a) image (b) vérité terrain (c) résultat d'un algorithme de segmentation

Quels critères pour décider que (b) et (c) sont similaires?

Comparer des formes binaires pour une segmentation

On s'intéresse au recouvrement des deux surfaces (Intersection Over Union=IOU)

- comparaisons ensemblistes: métrique de DICE, de jaccard

$$jaccard(A, B) = IoU(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

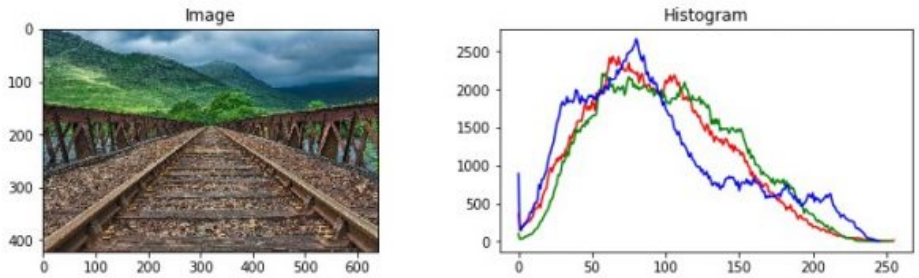
- Si on compare des entités de type *boite* (ex détecteur d'objet)

C'est le type de métrique utilisée pour entrainer **un détecteur d'objets**

Part III

Etude d'un descripteur conventionnel: l'histogramme

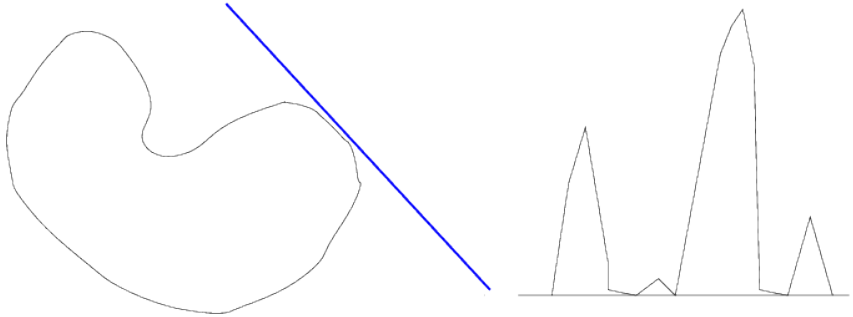
Utilisation des histogrammes comme descripteurs d'une forme ou d'une image



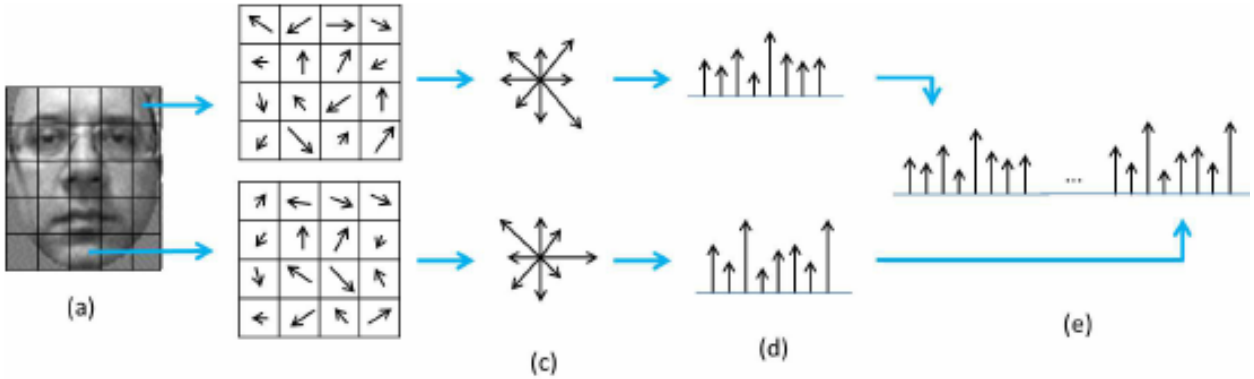
- un histogramme représente la répartition d'une variable continue à l'intérieur d'intervalles de valeurs appelés *bin* ou *baquets*.
- on peut représenter l'effectif d'un baquet ou représenter sa fréquence: l'histogramme est alors une représentation de la probabilité
- il permet une représentation **compacte** d'un objet
- représentation généralement non injective: plusieurs formes différentes partagent le même histogramme. Perte de l'information de spatialité.
- souvent utilisé pour constituer **rapidement** une short liste des formes candidates. Une étude plus fine est ensuite faite sur la short liste pour juger de la ressemblance. Voir par exemple [10]

Exemples de représentation avec des histogrammes

- **codage de la pente** Balayer un contour. Construire la distribution $f(x)$, ou l'histogramme de l'angle polaire de la tangente à la courbe.



- caractérisation de régions d'une image: HOG histogramme de gradient orienté [4]:



puis classification dans un SVM

Pour comparer des histogrammes

deux approches: comparer les **vecteurs** h ou comparer les **distributions** de probabilité associées:

Approche par comparaison de vecteurs:

- distance euclienne L_2 ou L_1 utilisables
- distance du χ^2

$$\chi^2(h_1, h_2) = \sum_{i=1}^{i=N} \frac{(h_1(i) - h_2(i))^2}{h_1(i) + h_2(i)}$$

- Les histogrammes sont souvent normalisés $\rightarrow$ (diviser h par $\sum_1^N h(i)$) . distance en cosinus: $1 - \sum h_1(i) * h_2(i)$ (deux vecteurs identiques normalisés ont un produit scalaire égal à 1)
- intersection des histogrammes $= 1 - \sum_{i=1}^{i=N} min(h_1(i), h_2(i)) / \sum_1^N h_1(i)$

Pour comparer des histogrammes

Approche par comparaison de vecteurs: Quelques remarques

- distance euclienne L_2 ou L_1 utilisables
  L_1 est plus robuste que L_2 à des erreurs
- distance du Chi 2:

$$\chi^2(h_1, h_2) = \sum_{i=1}^{i=N} \frac{(h_1(i) - h_2(i))^2}{h_1(i) + h_2(i)}$$

  l'erreur est pondérée par la hauteur du bin: un même écart est jugé moins important si h(i) est grand

- intersection des histogrammes $= 1 - \sum_{i=1}^{i=N} min(h_1(i), h_2(i)) / \sum_1^N h_1(i)$
  les endroits où $h_1(i)$ *ou* $h_2(i) = 0$ **ne sont pas considérés**

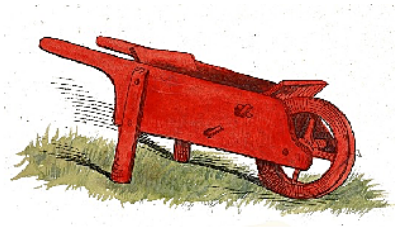
Comparer des histogrammes via les probabilités

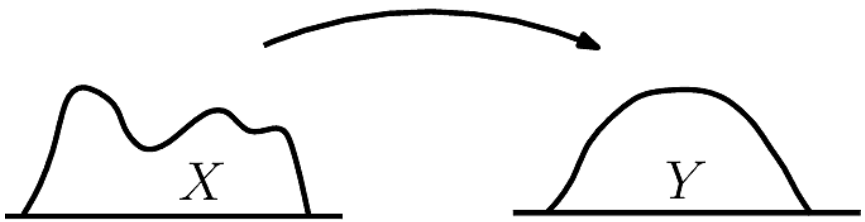
On considère des histogrammes normalisés, interprétés comme une distribution de probabilité:

- divergence de Kullback-Leibler entre distributions de probabilité

$$KL(p,q) = \sum_i p(i) log \frac{p(i)}{q(i)}$$

- distance du terrassier (ou Wasserstein) [11](métaphore du travail(= $\sum$poids*distance) **minimum** qu'un cantonnier doit fournir pour transformer un tas de terre en un autre)



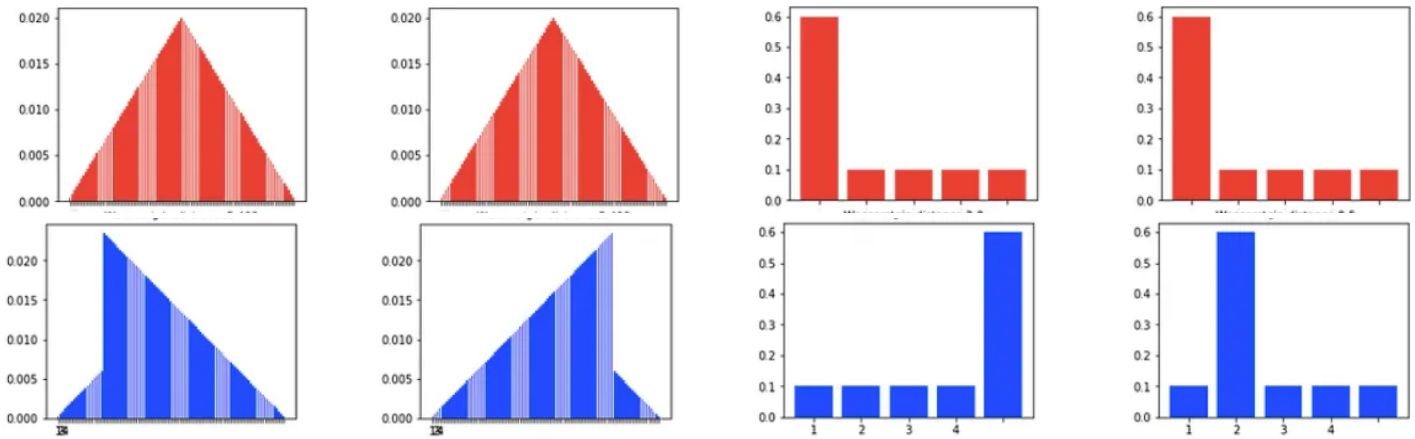


Voir la page terrassier. Un problème de transport optimal. **Permet de comparer des distribs qui n'ont pas le même support**

voir [1] pour une revue des métriques utilisables pour les histogrammes

Comparaison KL et Terrassier

Voici 4 exemples de distributions que l'on souhaite comparer. Que donneront respectivement les distances de KL et du terrassier sur ces exemples?



Exemple de calcul de distance

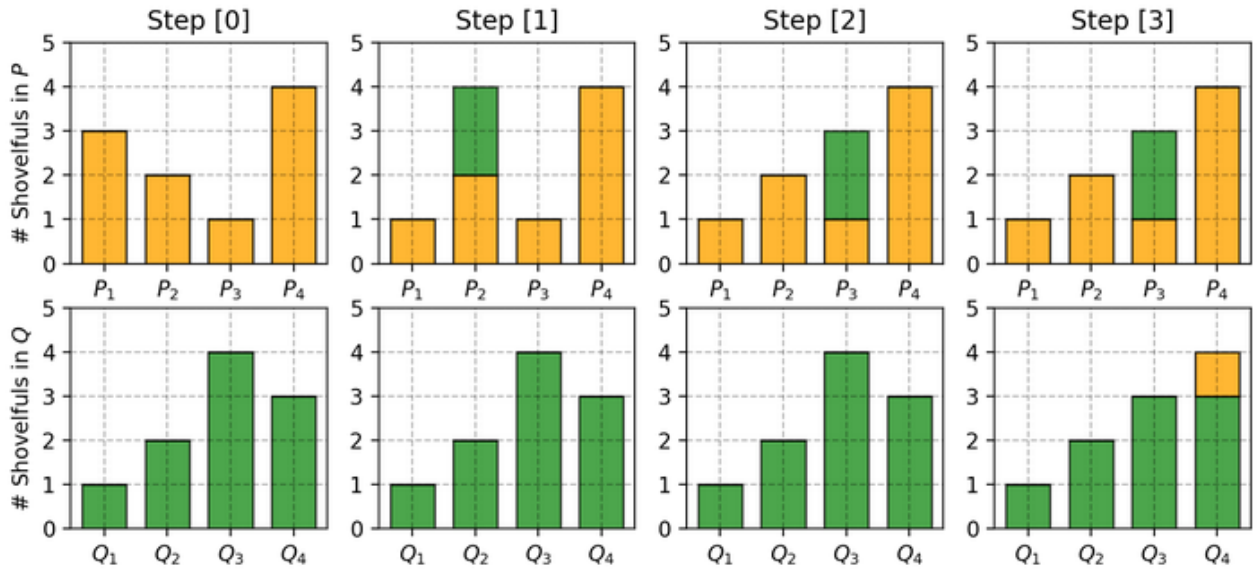
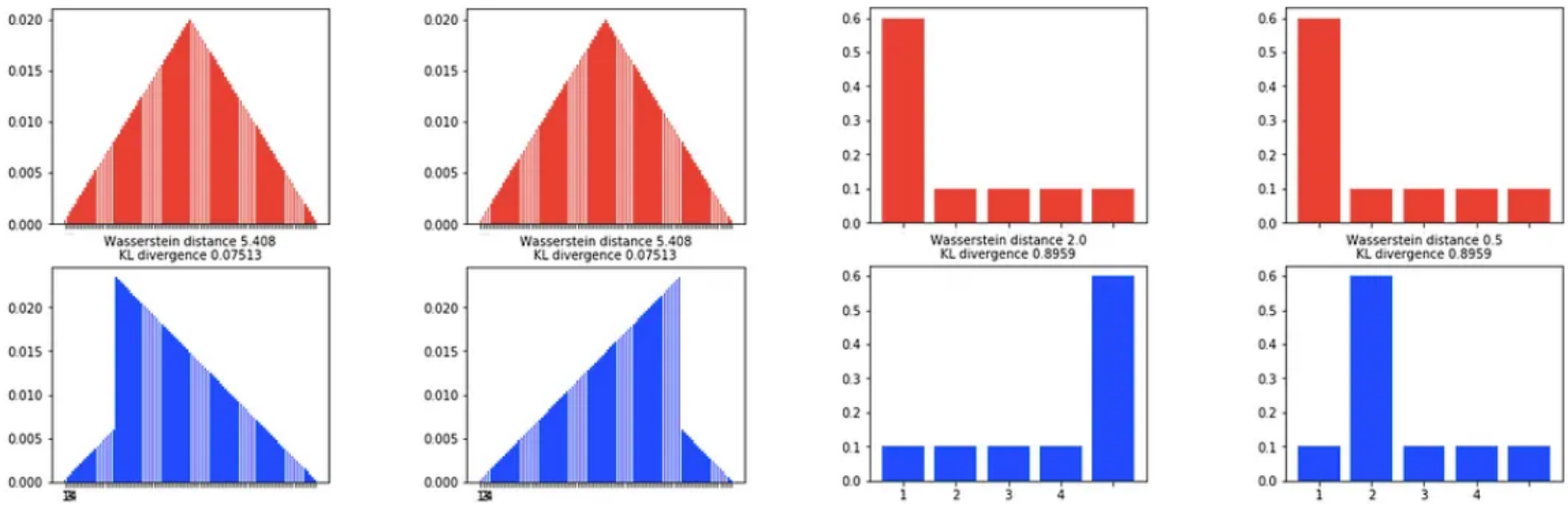


Figure 7: Step-by-step plan of moving dirt between piles in P and Q to make them match.

Distance= 2+2+1=5 Exemple tiré du site

Comparaison KL et Terrassier

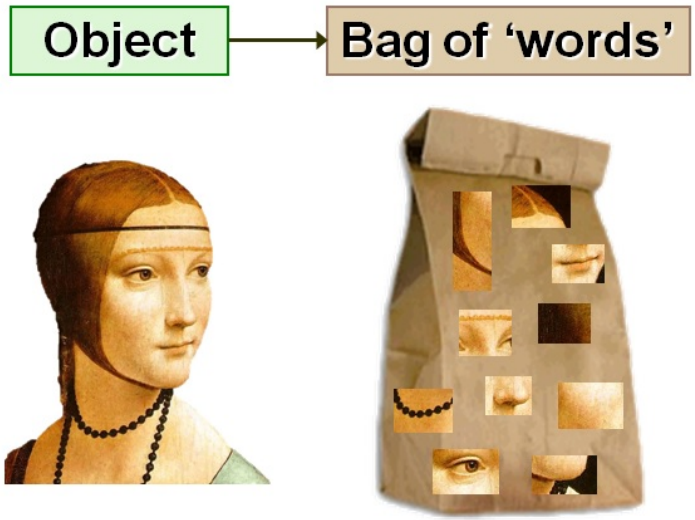
Voici 4 exemples de distributions que l'on souhaite comparer. Que donneront respectivement les distances de KL et du terrassier sur ces exemples?



qualités et défauts du Terrassier?
Voir par exemple Wasserstein generative adversarial networks; M. Arjovsky, L. Bottou

Représentation par sac de mots (Bag of words)

- Utiliser ou définir un vocabulaire. Définir un **vocabulaire visuel**
- analogie avec la comparaison de textes: on se base sur la fréquence d'apparition des mots
- comment définir un mot? $\rightarrow$ Par apprentissage
  - extraire des indices (keypoint,...) dans un grand ensemble d'images
  - classifier ces indices et **retenir comme mots les centres des classes**
- Représenter une image par un histogramme des mots visuels



- un outil très répandu
- la spatialité des mots n'est pas prise en compte dans la comparison (c'est bien un "sac" de mots)

- voir l'article initial de G. Csurka sur les sacs de mots [3] ainsi que Videogoogle [13].

Part IV

Apprendre des représentations et des métriques de similarité

De nouveaux descripteurs appris grâce aux réseaux convolutionnels

Dans l'approche par réseaux convolutionnels (CNN), extraction des caractéristiques et entraînement du classifieur **ne sont pas dissociés**:

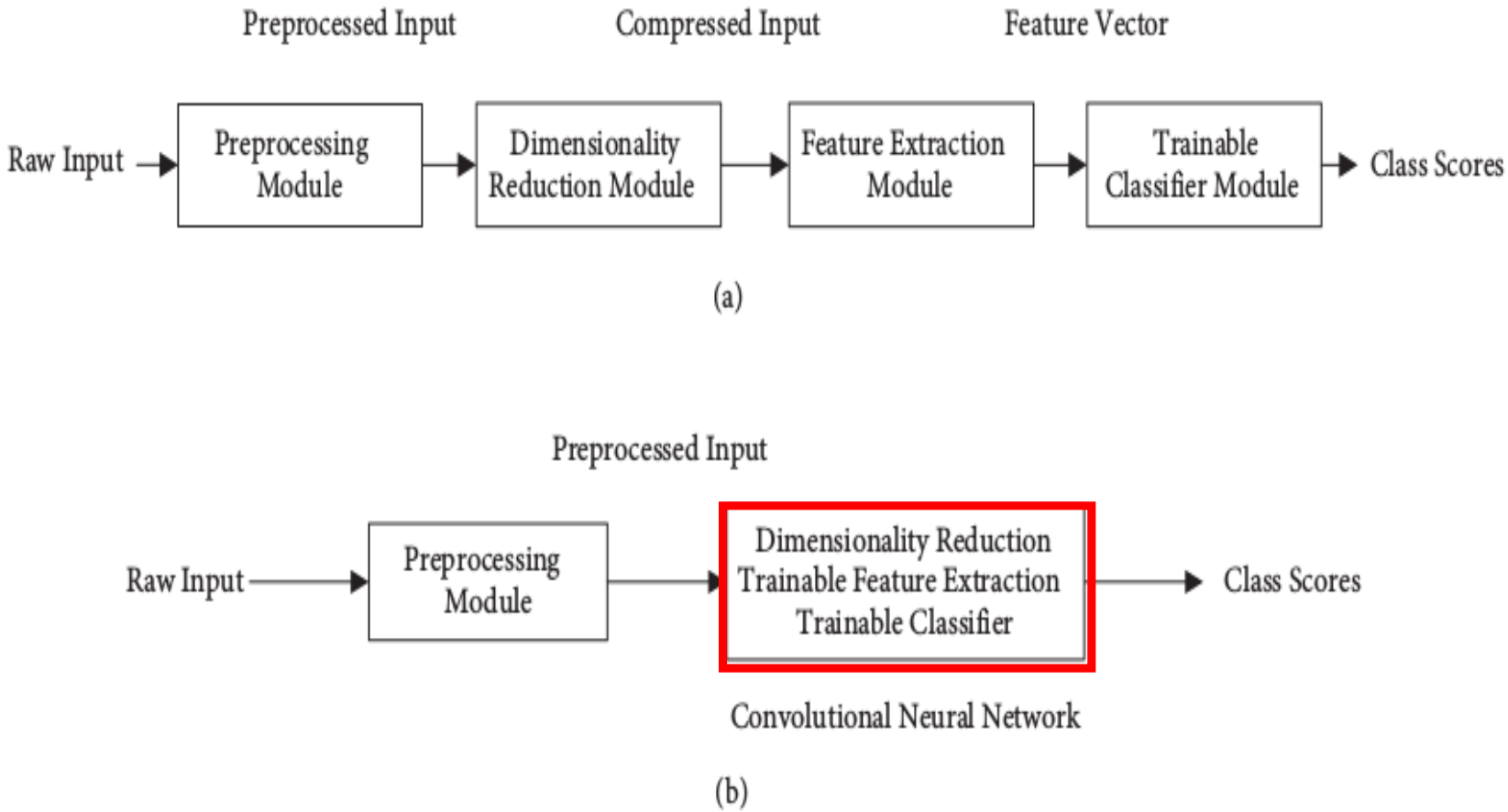


Figure 1. Pattern recognition approaches: (a) conventional, (b) CNN-based.

Un réseau convolutionnel

Exemple d'un des premiers réseaux convolutionnels pour la classification des chiffres: [9]

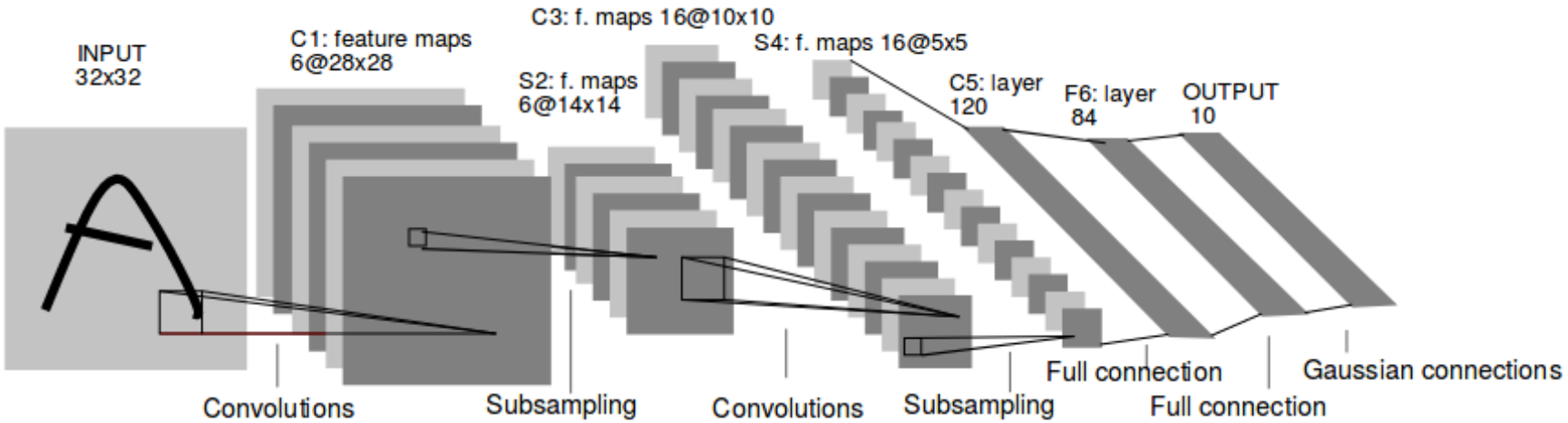


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Un réseau convolutionnel

Une succession de couches: on pourra consulter cette page de Stanford qui est un pense bête sur la structure des réseaux de neurones convolutionnels

- couches de **convolution** pour extraire des caractéristiques locales des images, à différentes échelles
- couche de **pooling**: réduire la taille tout en préservant les informations les plus importantes (average ou max pooling: garder la valeur maximale d'une fenêtre 4x4)
- couches Relu: fonction d'activation visant à introduire des non-linéarités
- couche **entièrement connectée** (FC): elle s'aplique sur une entrée "aplatie", donc à la fin des architectures de CNN. Chaque entrée est connectées à tous les neurones. Typiquement utilisée pour calculer les scores de classe.

Voir aussi le cours d'E. Vincent et J. Fix

Entraînement d'un réseau

Exemple de l'apprentissage supervisé:

- On dispose d'un ensemble de données $\{z_i\}$ dont la classe d_i (vérité terrain, étiquette) est **connue**
- Soit θ les paramètres ajustables du système (convolution,biais...)
- Pour une entrée z_i, le réseau calcule une valeur $f(z_i, \theta)$ qui prédit l'étiquette de z_i en sortie
- Les paramètres θ du système sont appris en minimisant la "loss' $E_{train}(\theta) = \sum_i dist(d_i, f_\theta(z_i))$, c'est-à-dire **l'écart entre données prédites et donnéees attendues** sur la base de donnée des exemples disponibles pour l'apprentissage.

Quelques "loss" classiques utilisées en apprentissage

Remarque: on peut définir de multiples loss dont le minimum est la solution souhaitée. Cependant la **convergence** vers le minimum peut être plus ou moins efficace (cf problème de la présence de "petits gradients" dans le cours d'estimation)

- régression: la sortie est un réel (ou un vecteur de réels). Si y_i est la vérité terrain correspondant à z_i la loss est

$$\sum_i ||y_i - f_\theta(z_i)||^2$$

  Si des données erronées existent utiliser L_1 ou L_{Huber} (voir le cours sur l'estimation robuste)
- classification multi-classes. Les sorties sont des probas d'appartenir à chacune des classes. On compare des *distributions*, souvent avec l'entropie croisée. Si y est la vérité terrain et $p = f_\theta(z)$, alors le coût pour un exemple est

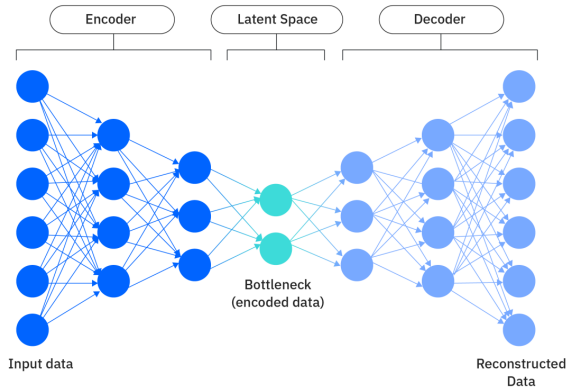
$$-\sum_c y_c log(p_c)$$

.

Pour plus de détails, reportez vous au livre [5] Deep Learning de Ian Goodfellow and Yoshua Bengio and Aaron Courville.

Espace latent et embedding

Un réseau passe par une représentation de petite taille avant de procéder à une tâche (regression calcul de distance, classification). Cette représentation est appelée **embedding** (ou plongement, ou descripteur convnet) et l'ensemble des représentations est l'**espace latent**.



- La dernière couche des réseaux avant FC a permis de bien classifier les données! C'est donc a priori un bon candidat pour décrire une forme
- des descripteurs *génériques*[1] proviennent de réseaux appris sur des grandes base de données Imagenet Large Scale Visual recognition challenge. Alexnet [8], googleLeNet [15], resNet [6]... sont des réseaux couramment utilisés pour produire ces descripteurs

[1] C'est la taille et la diversité des bases de données d'apprentissage qui rend d'une certaine façon le descripteur générique

Utilisation en reconnaissance de lieux

Voir Sunderhof [14]: reconnaissance de lieux malgré des points de vue très différents (pour la fermeture de boucle)

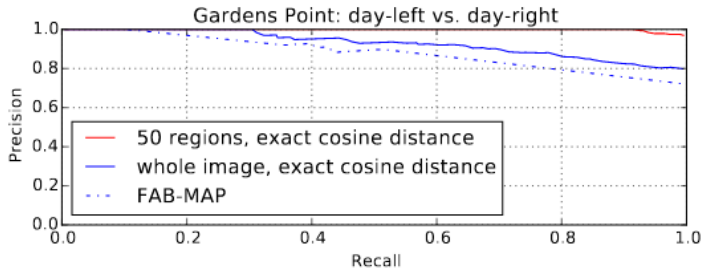


Fig. 3: Results for the Gardens Point Campus dataset. We clearly outperform the whole-image ConvNet-based method proposed by [41] and OpenFABMAP [14].

Fig. 4: Two example scenes from the Gardens Point Campus dataset with extracted and matched ConvNet landmarks. Notice the lateral camera displacement of several meters.

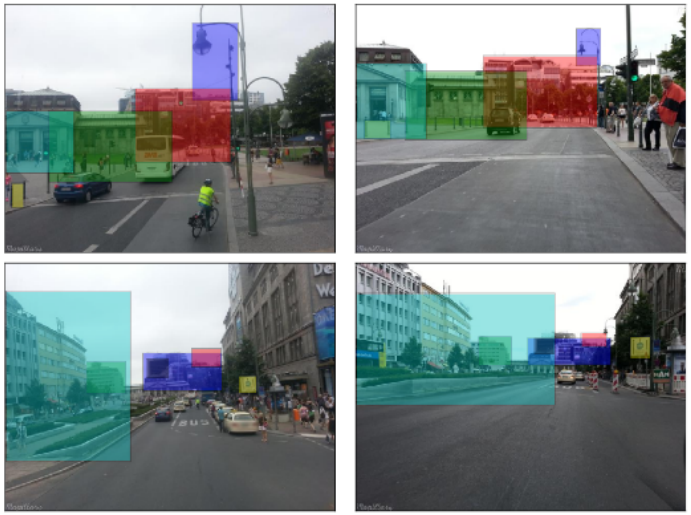


Fig. 5: Examples of successfully matched scenes from the *Berlin Kurfürstendamm* sequence of the Mapillary dataset. Images in a row belong to the same place but have been taken from different viewpoints, i.e. from the bike lane and from the upper deck of a tourist bus. The colored boxes illustrate some of the extracted and correctly matched landmarks.

Reconnaissance de lieux

Couplage

- d'une méthode de *box proposal*(Edge Box [16])
  - afin d'eviter de considérer toutes les boites possibles (sliding window)
  - repérer les fenêtres **intéressantes** comme étant celles qui contiennent un objet relativement isolé:
    critère:nombre de contours dans la boite - nombre de ces contours qui existent à l'extérieur de la boite
- de mise en correspondance entre ces boites utilisant la ressemblance des descripteurs CNN

Quelques idées sur EdgeBox

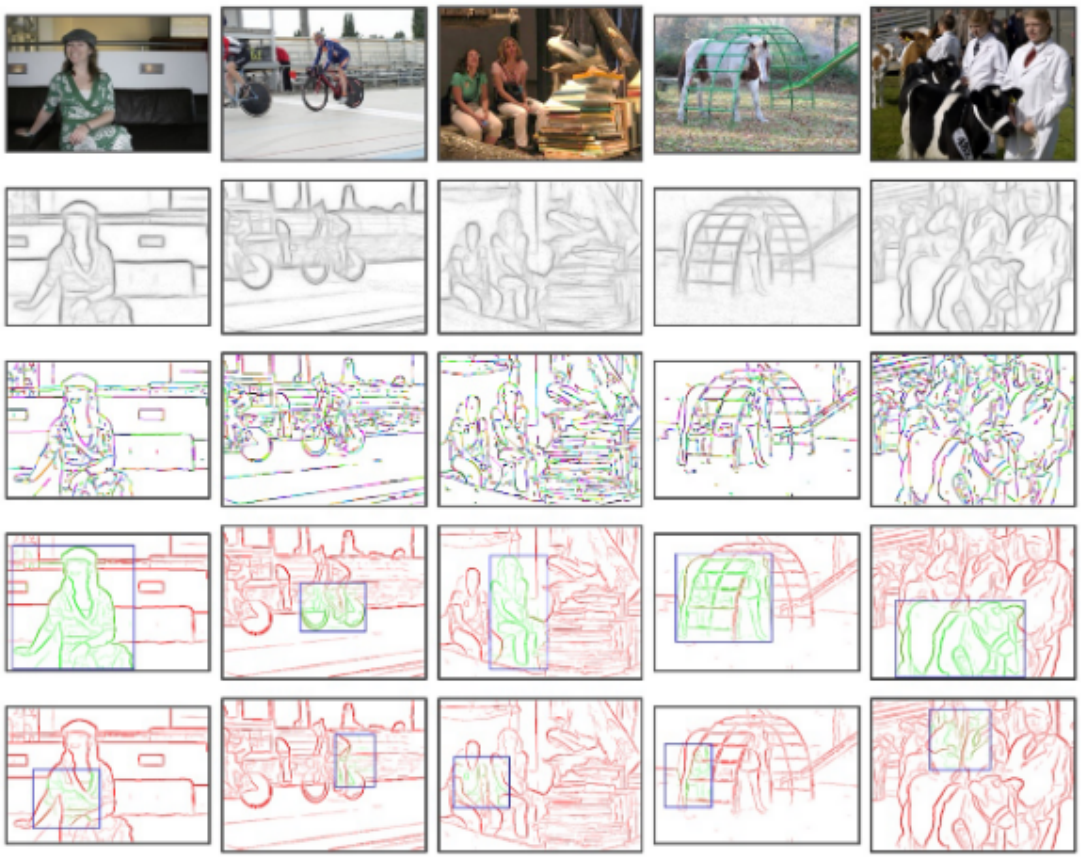


Fig. 1. Illustrative examples showing from top to bottom (first row) original image, (second row) Structured Edges [16], (third row) edge groups, (fourth row) example correct bounding box and edge labeling, and (fifth row) example incorrect boxes and edge labeling. Green edges are predicted to be part of the object in the box ($w_b(s_i) = 1$), while red edges are not ($w_b(s_i) = 0$). Scoring a candidate box based solely on the number of contours it *wholly encloses* creates a surprisingly effective object proposal measure. The edges in rows 3-5 are thresholded and widened to increase visibility.

Résultats

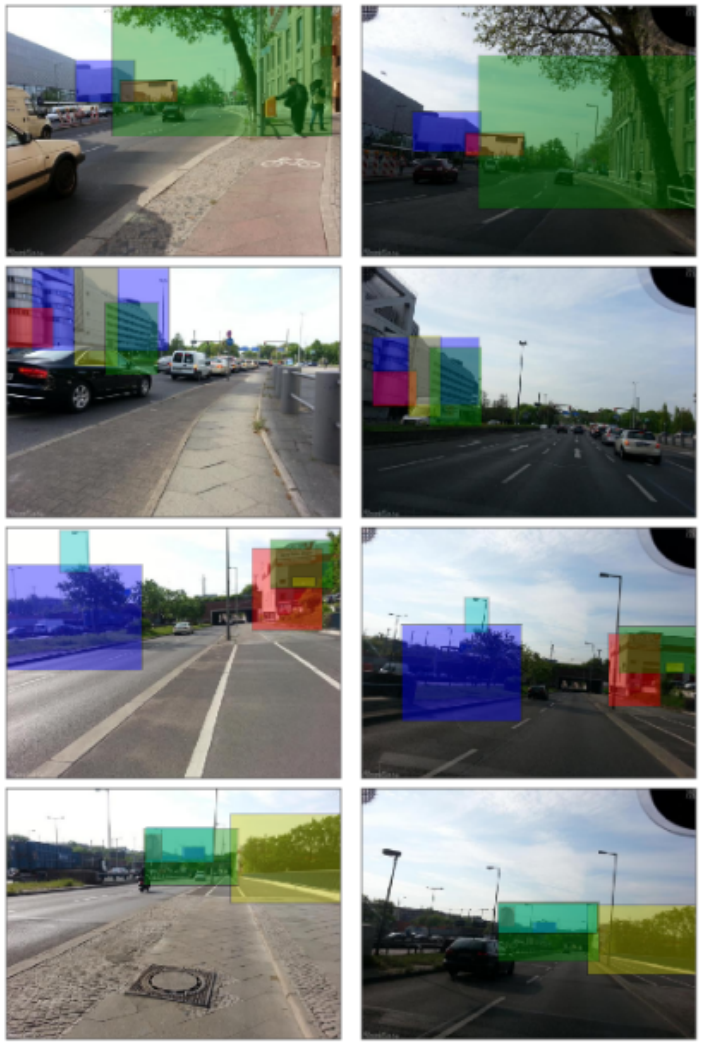
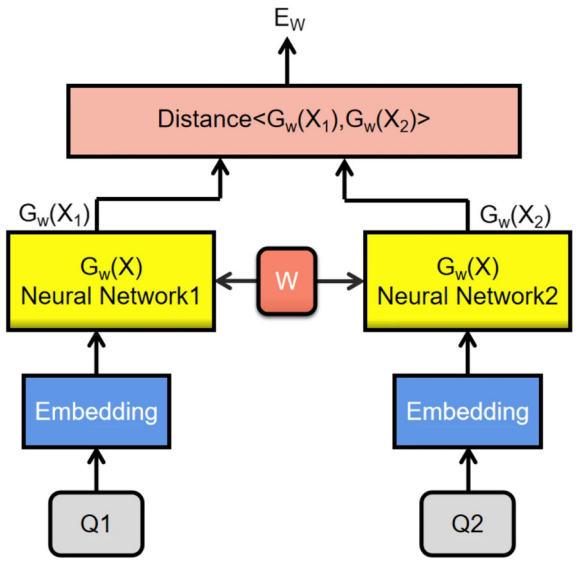


Fig. 6: The images in the *Berlin Halenseestraße* sequence have been recorded by a biker riding on the bike lane (left column) and a dashboard camera in the front of a car (right column). The changes in viewpoint are severe but our proposed method is able to extract landmarks and correctly match them between a large number of scenes.

L'apprentissage contrastif: Apprendre des descripteurs dédiés à une tâche, avec des données peu supervisées I

Avec les descripteurs convNet, on utilise un descripteur issu de la classification avec **des données supervisées**, ce qui est couteux.
Avec l'apprentissage constratif :

- On veut trouver des représentations (embeddings) en fournissant seulement des d'exemples positifs (qui se ressemblent) et des couples négatifs (qui sont dissemblables)
- Idée: apprendre un descripteur G_w de façon que les **exemples positifs** aient des **descripteurs proches** et que les **exemples négatifs** aient des **descripteurs éloignés**



Un réseau siamois [2]:

Pourquoi réduit-on la dépendance aux données supervisées?

- on peut utiliser des augmentations pour générer des exemples positifs (rotations, scale, déformations... d'une image)
- On peut aussi enlever un morceau d'une donnée et chercher à la prédire
- on peut générer des exemples synthétiques
- on peut utiliser des étiquetages plus light: des images prises d'un même lieu...
- on est moins dépendant des *class imbalance*
- et on peut aussi y inclure des données supervisées (prendre tous les exemples appartenant à une même classe de chiffres)

Conclusion

descripteur ou métrique fait main (handcrafted) ou appris?

- les CNNs ont apporté des progrès indéniables en terme d'invariance aux conditions environnementales (lumière, point de vue,...)
- l'utilisation des CNNs dépend de la disponibilité de données d'apprentissage. Pour pallier au manque de données:
  - re-training d'un réseau existant avec de nouvelles données
  - augmentation de données
  - utilisation de données synthétiques réalistes comme complément
- ces descripteurs ne supplantent pas forcément les descripteurs faits main. Voir par exemple: *Comparative Evaluation of Hand-Crafted and Learned Local Features* [12]
- il existe de nombreuses approaches hybrides mixant CNN et conventionnel

Annexe: Etude de l'article Video Google

- Quel est le but de l'article?
- Quelle est l'analogie faite par les auteurs entre le problème considéré et le "text retrieval"?
- Qu'est ce que le vocabulaire visuel? Comment est-il construit?
- Comment est construit le descripteur global d'une image?
- On considère deux images contenant les mêmes objets mais ceux ci sont placés à des endroits différents dans l'image. Les deux images auront-t-elles un descripteur global similaire?
- Pourquoi y-a-t-il besoin d'imposer des contraintes de spatialité dans la mise en correspondance?
- A quoi sert la stop liste?

Bibliographie I

[1] Sung-Hyuk Cha and Sargur N. Srihari. "On measuring the distance between histograms". In: *Pattern Recognition* 35.6 (2002), pp. 1355–1370. ISSN: 0031-3203. DOI: https://doi.org/10.1016/S0031-3203(01)00118-2. URL: http://www.sciencedirect.com/science/article/pii/S0031320301001182.

[2] Sumit Chopra, Raia Hadsell, and Yann LeCun. "Learning a similarity metric discriminatively, with application to face verification". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE. 2005, pp. 539–546.

[3] G. Csurka et al. "Visual categorization with bags of keypoints". In: *Workshop on Statistical Learning in Computer Vision, ECCV*. 2004, pp. 1–22.

[4] Navneet Dalal and Bill Triggs. "Histograms of Oriented Gradients for Human Detection". In: *In CVPR*. 2005, pp. 886–893.

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

Bibliographie II

[6] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016. DOI: 10.1109/CVPR.2016.90.

[7] Anil Jain, Arun Ross, and S. Pankanti. "Biometrics: a tool for information security. IEEE Tran Inform Forensics Secur". In: *Information Forensics and Security, IEEE Transactions on* 1 (July 2006), pp. 125–143. DOI: 10.1109/TIFS.2006.873653.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'12. Lake Tahoe, Nevada: Curran Associates Inc., 2012, pp. 1097–1105. URL: http://dl.acm.org/citation.cfm?id=2999134.2999257.

[9] Yann Lecun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE*. 1998, pp. 2278–2324.

Bibliographie III

[10] G. Mori, S. Belongie, and J. Malik. "Shape contexts enable efficient retrieval of similar shapes". In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 1. 2001. DOI: 10.1109/CVPR.2001.990547.

[11] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. "The Earth Mover's Distance As a Metric for Image Retrieval". In: *Int. J. Comput. Vision* 40.2 (Nov. 2000), pp. 99–121. ISSN: 0920-5691. DOI: 10.1023/A:1026543900054. URL: https://doi.org/10.1023/A:1026543900054.

[12] Johannes L. Schönberger et al. "Comparative Evaluation of Hand-Crafted and Learned Local Features". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6959–6968. DOI: 10.1109/CVPR.2017.736.

[13] Sivic and Zisserman. "Video Google: a text retrieval approach to object matching in videos". In: *Proceedings Ninth IEEE International Conference on Computer Vision*. Oct. 2003, 1470–1477 vol.2. DOI: 10.1109/ICCV.2003.1238663.

Bibliographie IV

[14] Niko Sünderhauf et al. "Place Recognition with ConvNet Landmarks: Viewpoint-Robust, Condition-Robust, Training-Free.". In: *Robotics: Science and Systems*. 2015. URL: http://dblp.uni-trier.de/db/conf/rss/rss2015.html#SunderhaufSJDPU15.

[15] Christian Szegedy et al. "Going Deeper with Convolutions". In: *CoRR* abs/1409.4842 (2014). arXiv: 1409.4842. URL: http://arxiv.org/abs/1409.4842.

[16] C. Lawrence Zitnick and Piotr Dollár. "Edge Boxes: Locating Object Proposals from Edges". In: *ECCV*. 2014.