

Images, pixels, statistique et formes

Cédric Beaulac

Université du Québec à Montréal

25 mai 2024

L'objectif de la présentation est d'introduire les fondations de l'analyse d'image, de l'analyse de données fonctionnelles ainsi que de construire un pont entre ces deux domaines.

Remerciement à mes collègues, les professeurs Fabrice Larribe (UQAM), Marie-Hélène Descary (UQAM) et Jiguo Cao (SFU) et à mes étudiants Louis-Pierre Ménard (UQAM), Mélanie Raymond (UQAM), Valentin Larchevêque (Montpellier) et Sidi Wu (SFU).

Images, pixels, statistique et formes

Images

Données fonctionnelles

Données fonctionnelles 2D

Formes et travaux à venir

Images

- ▶ Une image, c'est quoi ?
- ▶ On fait référence a une surface colorée deux-dimensions (2D)
- ▶ Les termes image et photo seront utilisées de manière interchangeable.



Figure: Image en couleur d'une pizza.

Pixels

Essayons maintenant de définir l'objet mathématique qu'on utilise pour représenter une image:

Une image de résolution (h, l) , de l pixels de large et h pixels de haut peut être représenté par une matrice de taille $h \times l$, disons I où le terme $I_{i,j}$ représente la coloration du pixel i, j .

Par simplicité, parlons d'image en teinte de gris pour commencer; cette coloration sera souvent identifiée par une valeur dans l'intervalle $[0, 255]$

Blanc=0, noir=255 et nous avons un spectre de couleurs pour les valeurs intermédiaires

Il y a donc $256 = 2^8$ valeurs possibles pour la coloration d'un pixel.



Figure: Image d'un '5' provenant de la base de données MNIST.

	[1,1]	[1,2]	[1,3]	[1,4]	[1,5]	[1,6]	[1,7]	[1,8]	[1,9]	[1,10]	[1,11]	[1,12]	[1,13]	[1,14]	[1,15]	[1,16]	[1,17]	[1,18]	[1,19]	[1,20]	[1,21]	[1,22]	[1,23]	[1,24]	[1,25]	[1,26]	[1,27]	[1,28]
[1,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[2,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[3,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[4,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[5,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[6,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[7,1]	0	0	0	0	0	0	0	0	30	36	94	154	170	253	253	253	253	253	225	172	253	242	195	64	0	0	0	0
[8,1]	0	0	0	0	0	0	0	49	238	253	253	253	253	253	253	253	253	251	91	82	82	56	39	0	0	0	0	0
[9,1]	0	0	0	0	0	0	0	18	219	253	253	253	253	253	188	182	247	241	0	0	0	0	0	0	0	0	0	0
[10,1]	0	0	0	0	0	0	0	80	155	107	253	253	205	11	0	45	154	0	0	0	0	0	0	0	0	0	0	0
[11,1]	0	0	0	0	0	0	0	24	1	154	253	90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[12,1]	0	0	0	0	0	0	0	0	0	0	139	253	190	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[13,1]	0	0	0	0	0	0	0	0	0	11	180	253	70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[14,1]	0	0	0	0	0	0	0	0	0	0	35	241	225	160	108	1	0	0	0	0	0	0	0	0	0	0	0	0
[15,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	81	240	253	253	119	25	0	0	0	0	0	0	0	0	0
[16,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	45	186	253	253	150	27	0	0	0	0	0	0	0	0
[17,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	93	252	253	187	0	0	0	0	0	0	0	0
[18,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	249	253	249	64	0	0	0	0	0	0	0
[19,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	46	130	183	253	253	207	2	0	0	0	0	0	0	0
[20,1]	0	0	0	0	0	0	0	0	0	0	39	148	229	253	253	253	250	182	0	0	0	0	0	0	0	0	0	0
[21,1]	0	0	0	0	0	0	0	0	24	114	221	253	253	253	253	201	78	0	0	0	0	0	0	0	0	0	0	0
[22,1]	0	0	0	0	0	0	0	0	23	86	213	253	253	253	198	81	2	0	0	0	0	0	0	0	0	0	0	0
[23,1]	0	0	0	0	0	0	0	18	171	219	253	253	253	195	80	9	0	0	0	0	0	0	0	0	0	0	0	0
[24,1]	0	0	0	0	0	0	0	55	172	226	253	253	253	244	113	11	0	0	0	0	0	0	0	0	0	0	0	0
[25,1]	0	0	0	0	0	0	0	116	253	253	253	212	135	132	16	0	0	0	0	0	0	0	0	0	0	0	0	0
[26,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[27,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[28,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure: Représentation matricielle de l'image du '5'.

On peut aussi représenter les images comme une surface où l'intensité de coloration (la surface) varie en fonction de la position (i, j) .

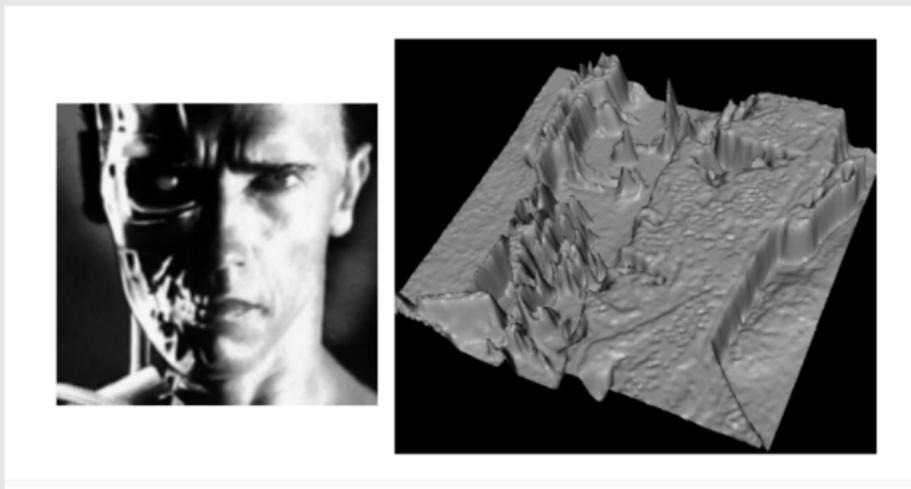


Figure: Une image à gauche et la surface qui en découle à droite.

La hauteur (z) est donc l'intensité $I_{i,j}$.

Et les images en couleurs elles ?

Tout comme le spectre de couleurs observables qui est une combinaison de couleurs, pour former un pixel coloré nous combinons du rouge (R), du bleu (B) et du vert (G).

Encore une nous identifions l'intensité de chaque couleur à l'aide de 8 bits, donc sur un spectre $[0, 255]$.

La couleur d'un pixel est représentée par une combinaison de RGB chacun sur $[0, 255]$.

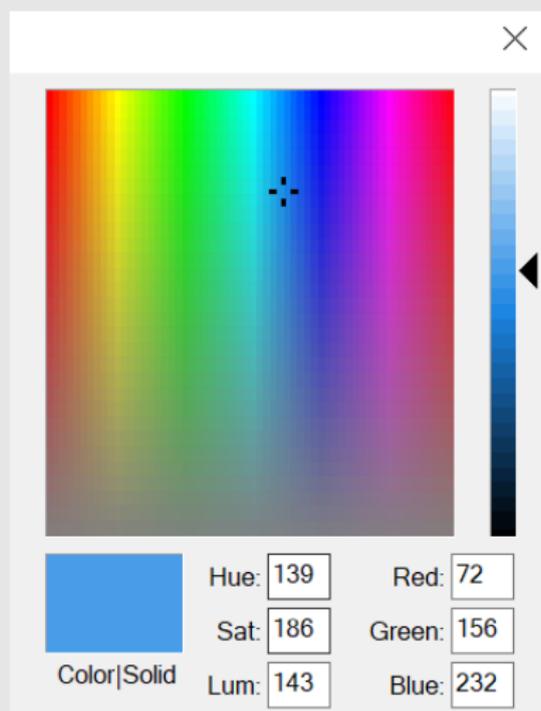


Figure: Représentation d'une couleur comme d'une combinaison *red*, *blue*, *green*.

Conséquemment, l'objet mathématique représentant une image couleur de l pixels de large et h pixels de haut est un tensor 3-dimensions de taille $(3, h, l)$.

Nous avons essentiellement une matrice de taille (h, l) pour chacune des trois couleurs.

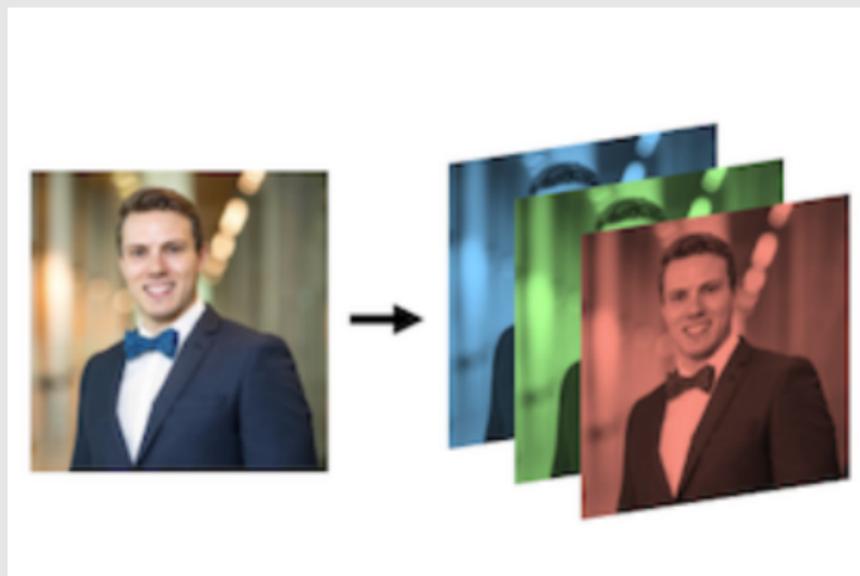


Figure: Une combinaison de 3 teintes forme une image en couleur.

État des choses en analyse d'image

Considérant que le pixel et la représentation matricielle est le format dans lequel les images sont enregistrées, les techniques d'analyse d'images, autant modernes qu'anciennes, furent développées pour ces objets.

État des choses en analyse d'image

C'est-à-dire que les outils pour **modifier**, **transformer** et **analyser** les images considèrent ceux-ci exclusivement comme une **collection de pixels**, enregistrée sous forme matricielle.

État des choses en analyse d'image

Par exemple, les modèles modernes d'apprentissage machine et d'intelligence artificielle moderne et les modèles génératifs d'image. Mais aussi les modèles plus anciens développés en géométrie digitale pour l'ajustement de contraste et la détection de contour.

Filtration

Une opération-pixel standard utilisé dans toutes les situations de modification, transformation et analyse d'image est la filtration.

Au centre de la filtration, l'opération de convolution.

Filtration

Soit $*$ l'opérateur de la convolution 2-dimensions, il s'agit d'une combinaison linéaire entre une sous-matrice (un voisinage de l'image) et un noyau de convolution (un filtre).

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i + m, j + n) K(m, n)$$

On glisse le noyau K sur la matrice I et on obtient une nouvelle matrice S .

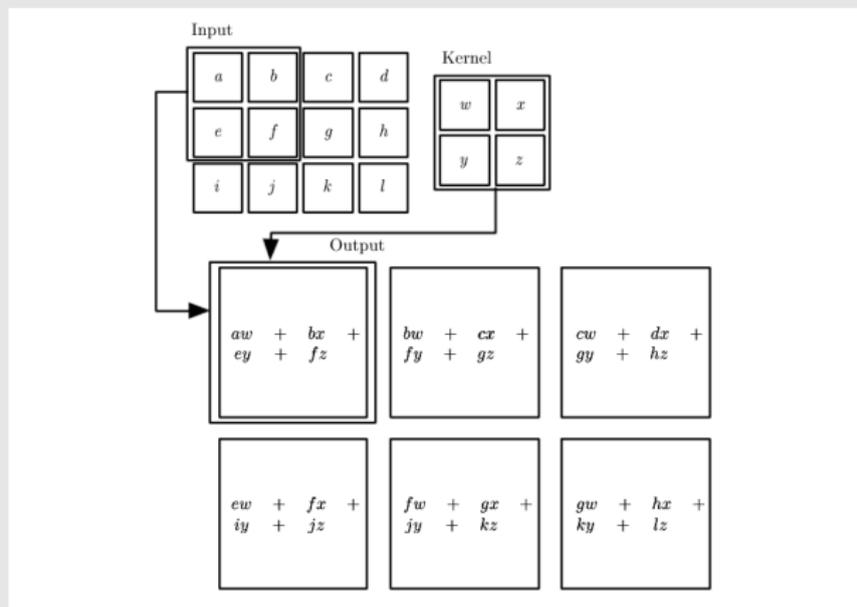
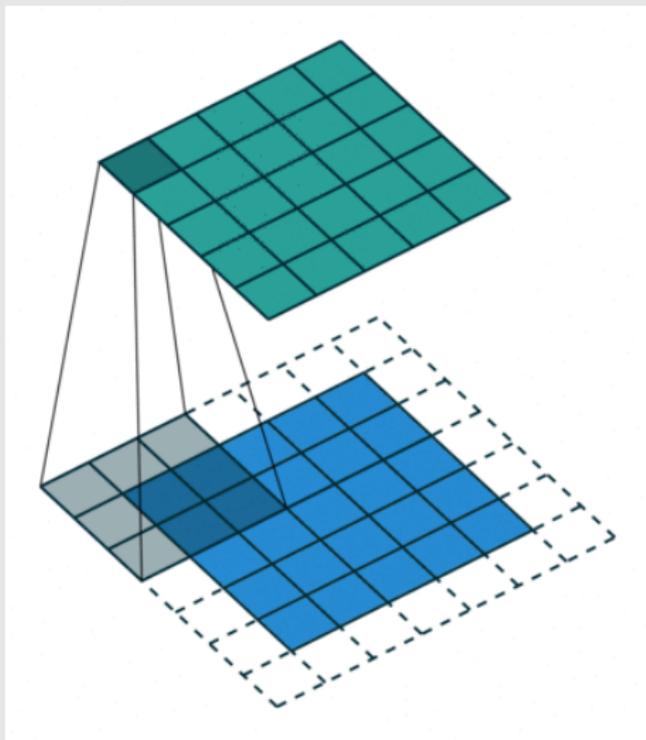
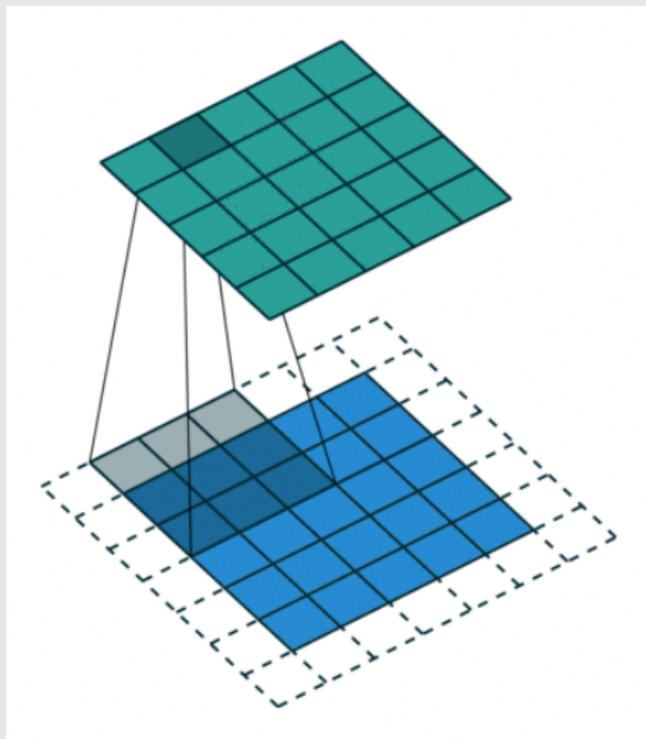
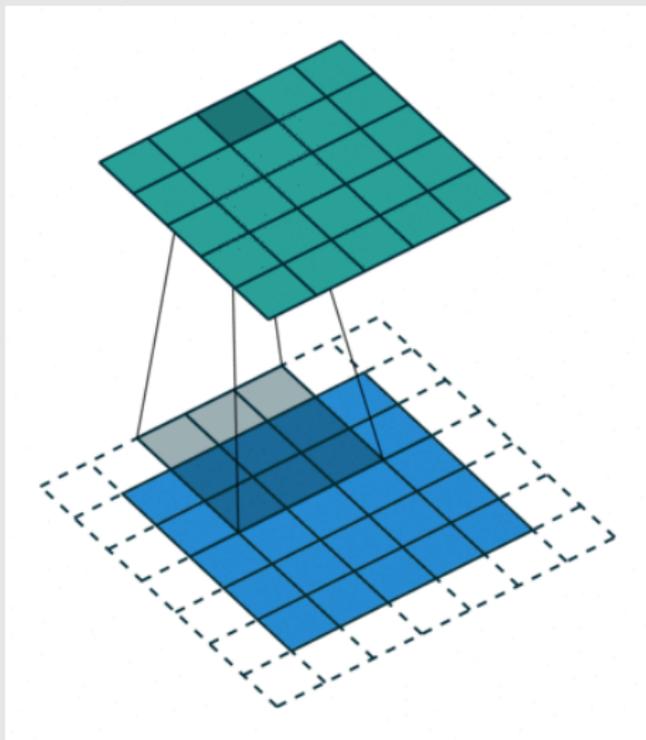
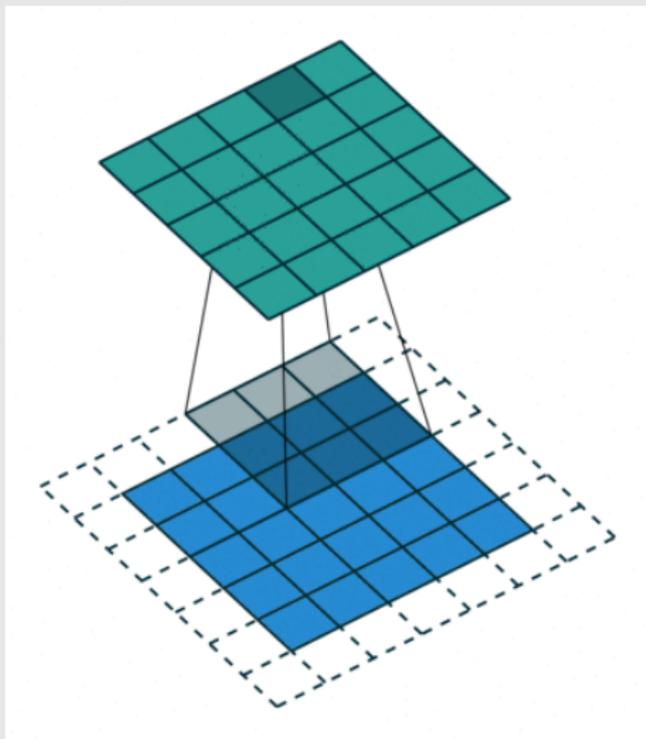


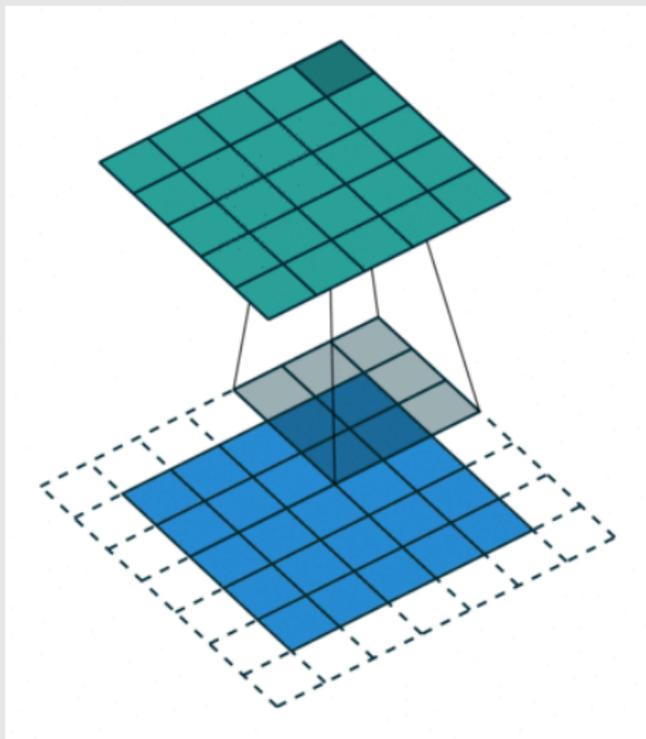
Figure: Application de la convolution entre l'entrée (3X4) (Input) et le filtre (Kernel). Le résultat est de dimension 2X3

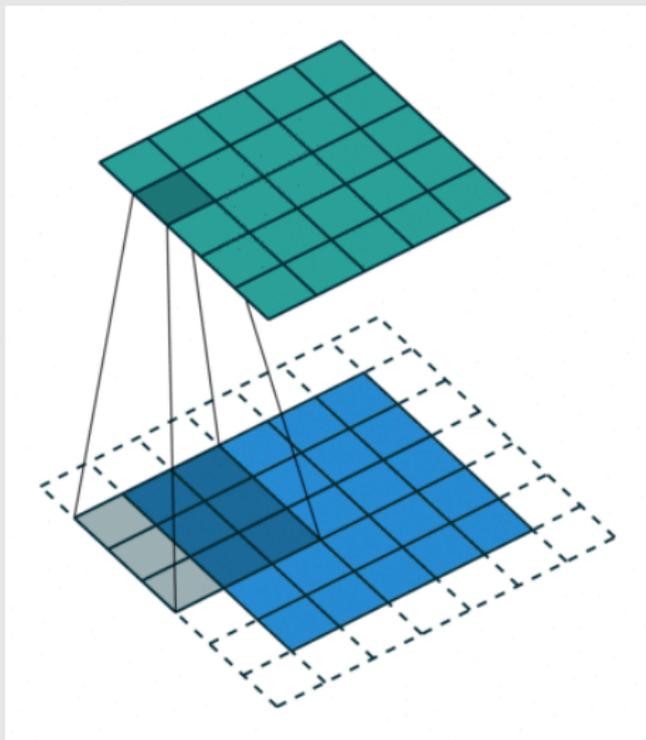


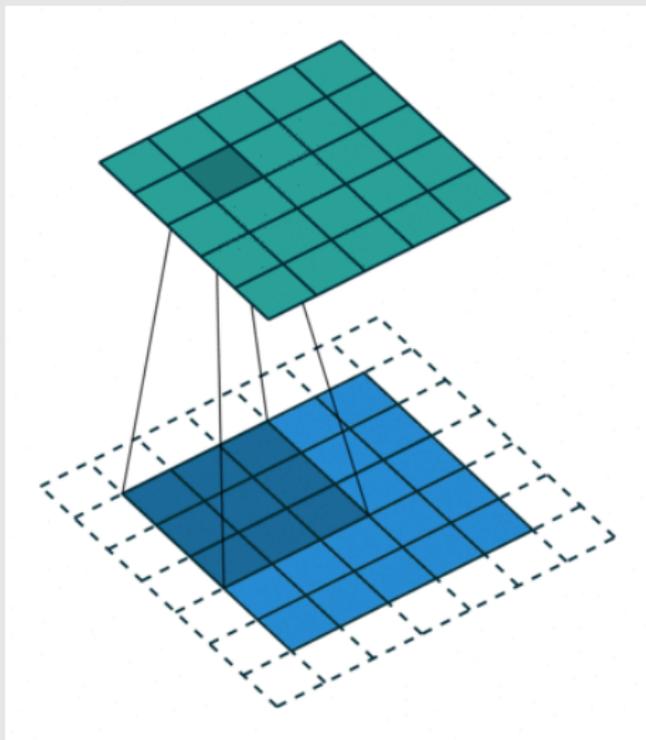


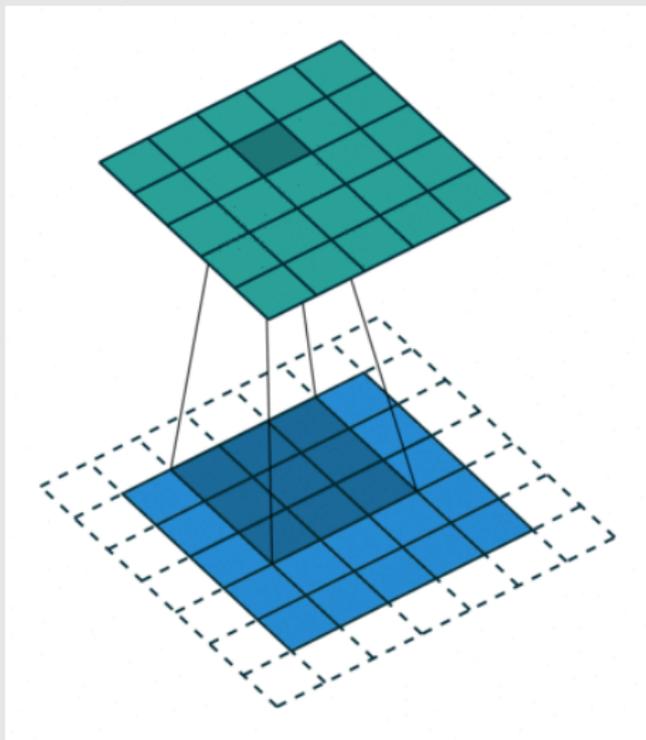


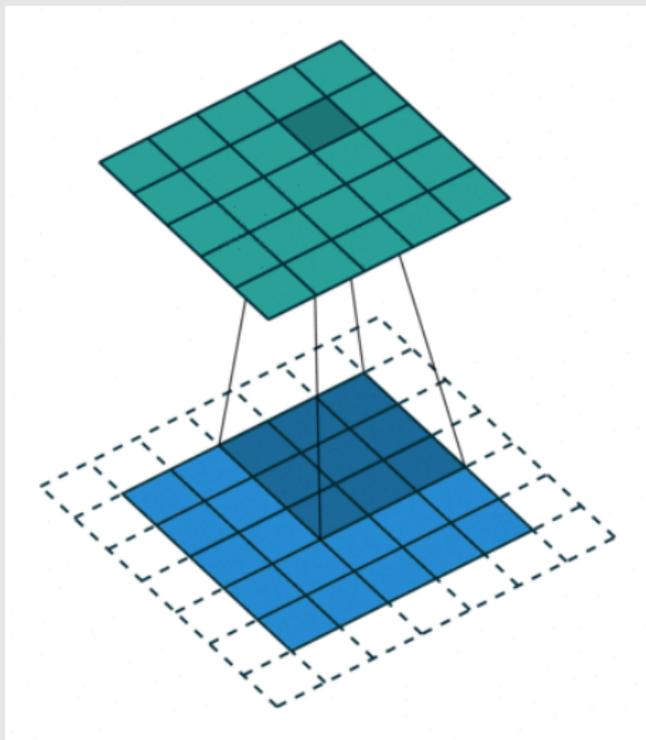


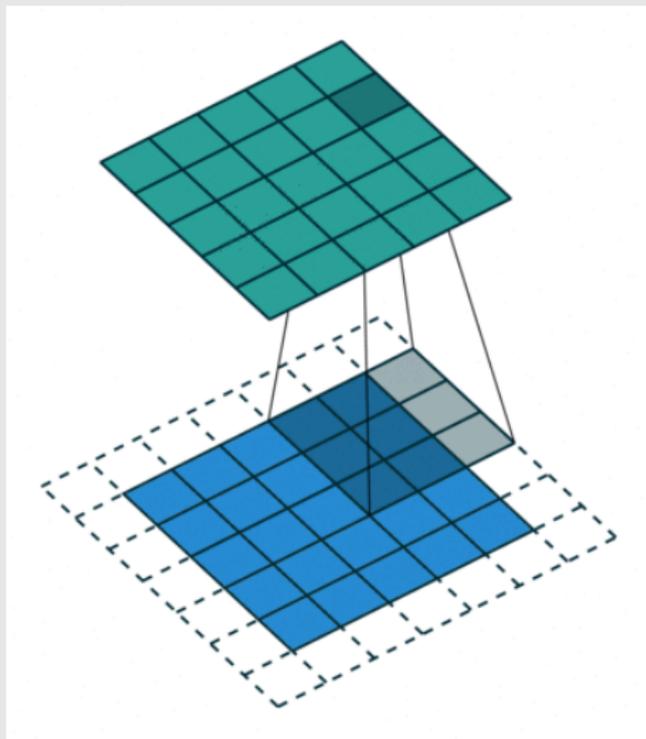


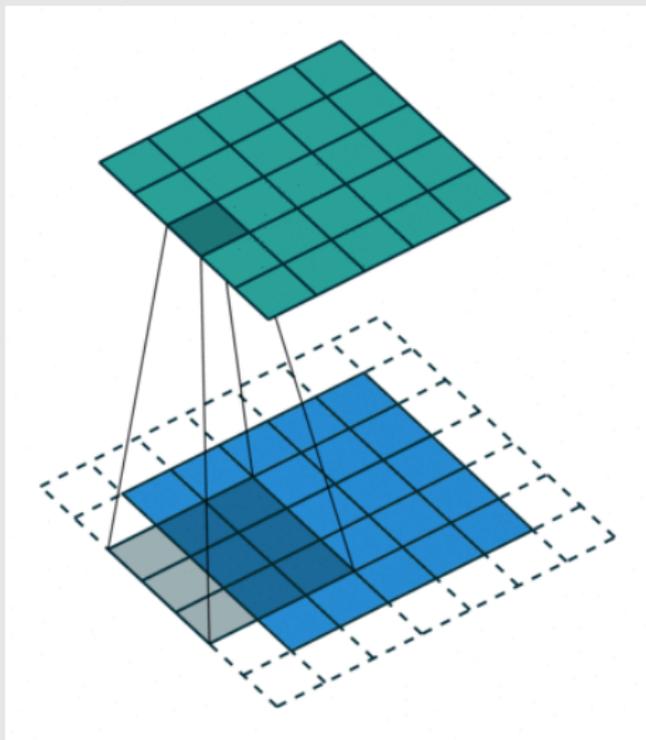


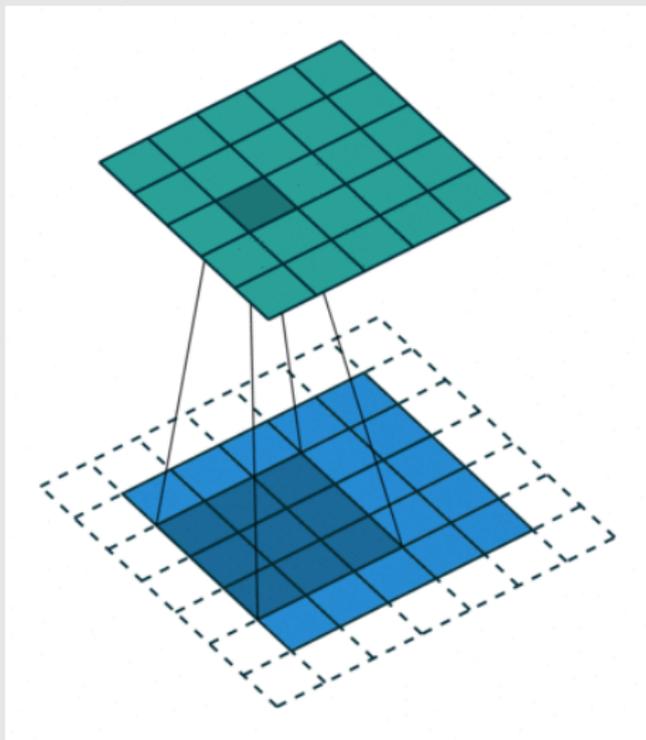


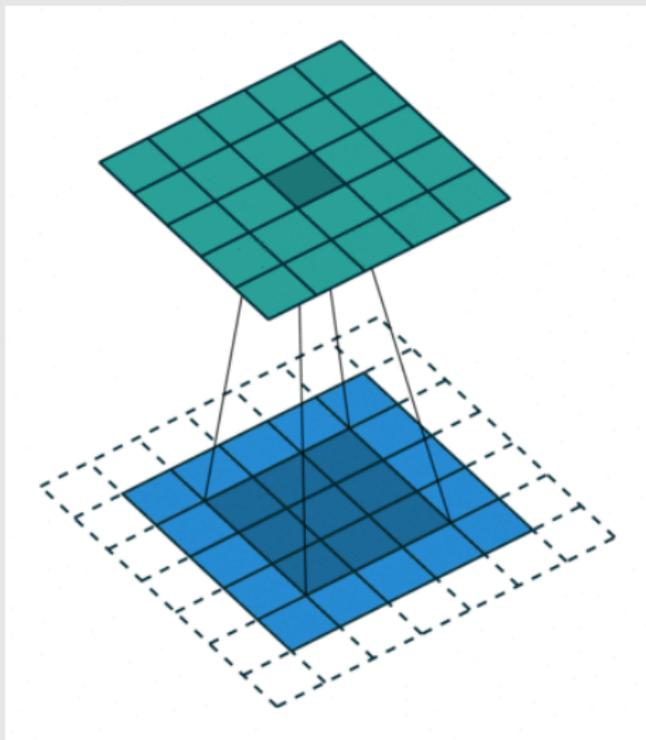


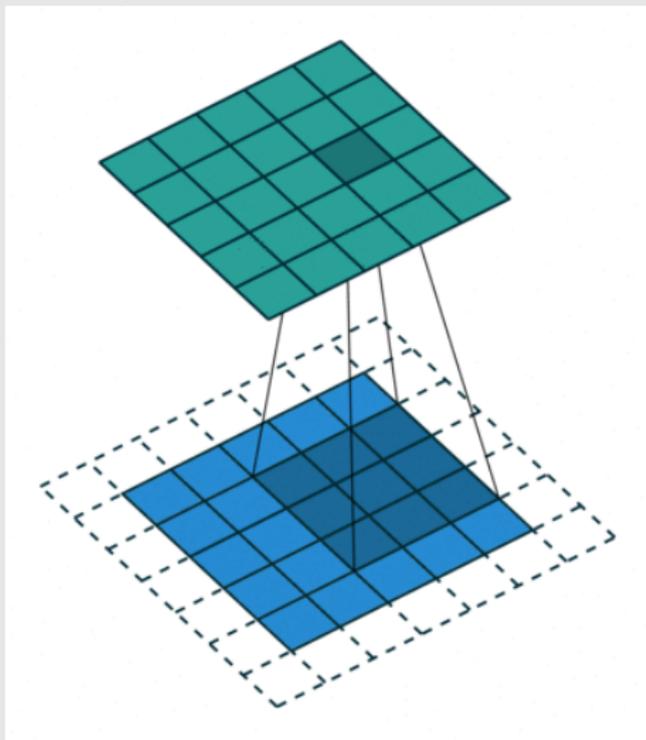


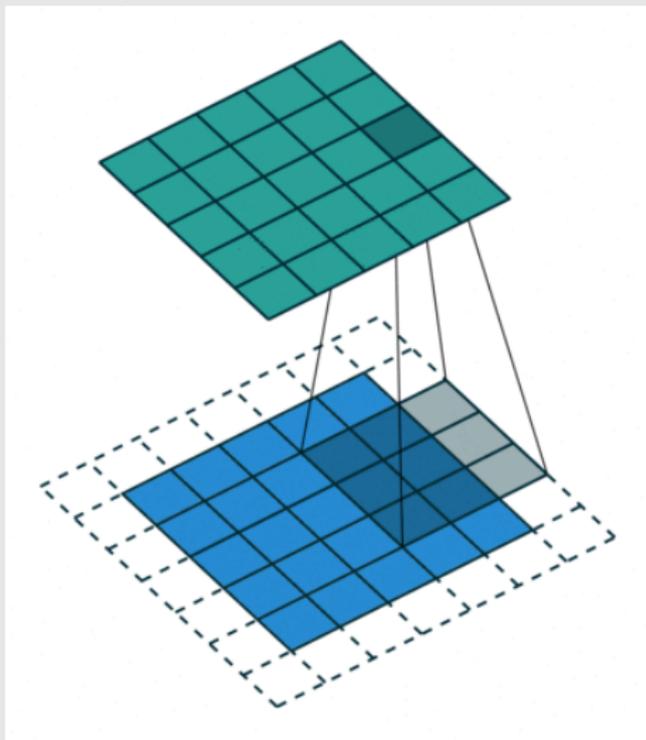


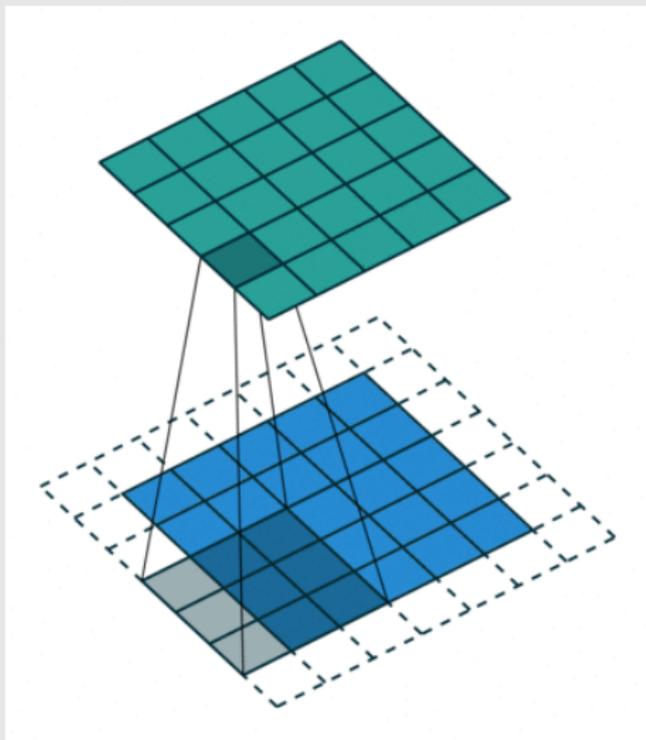


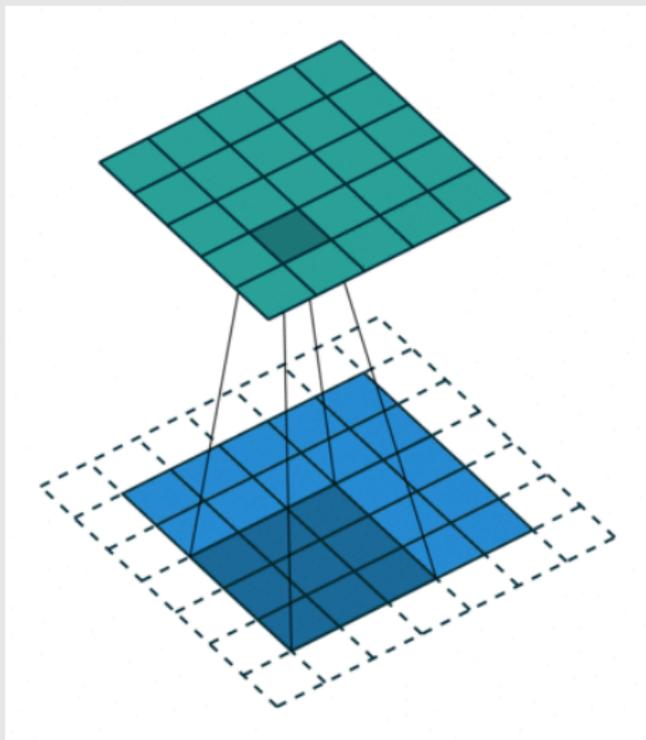


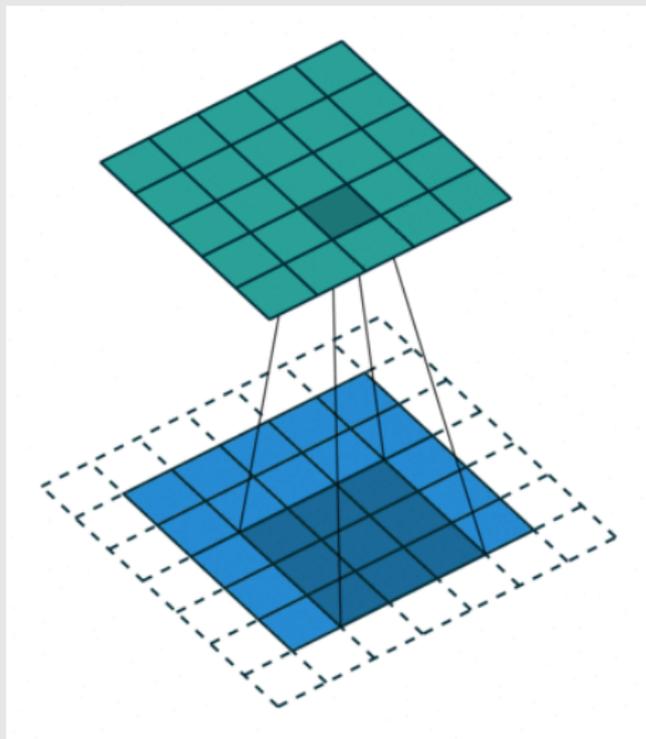


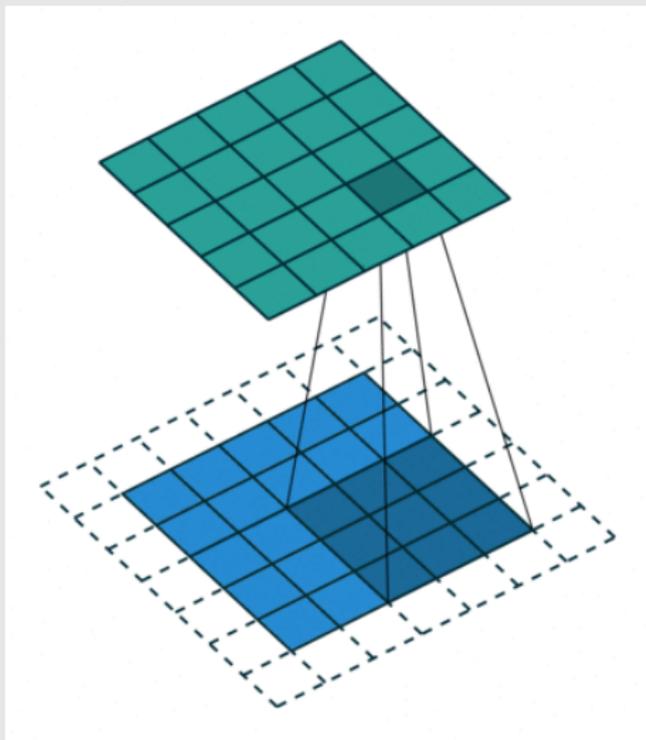


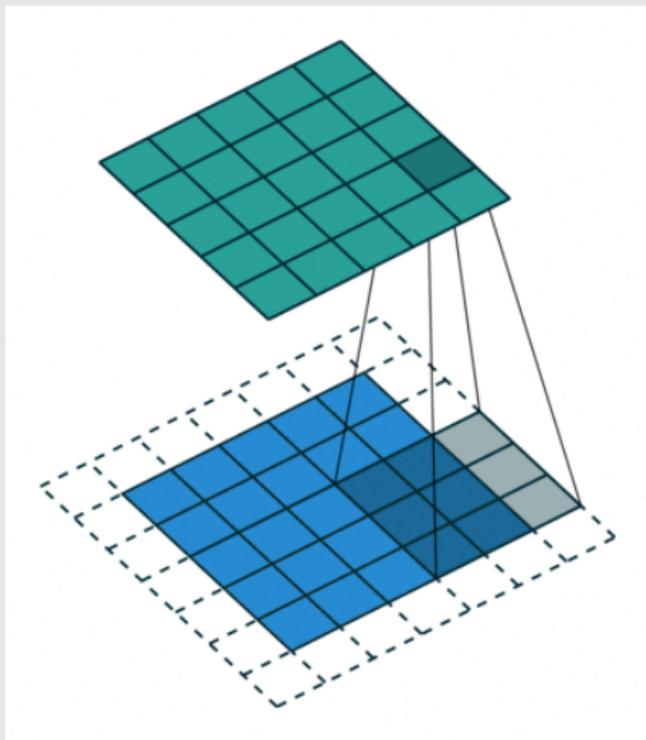


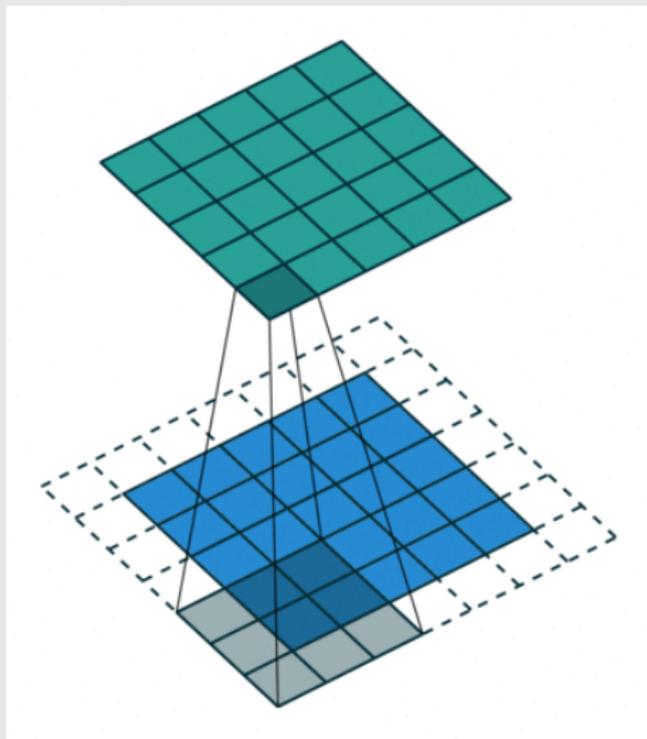


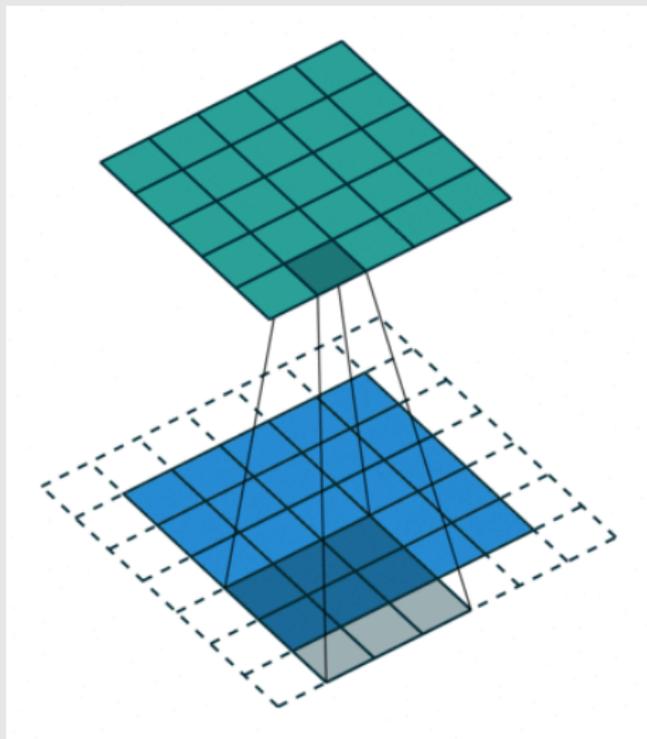


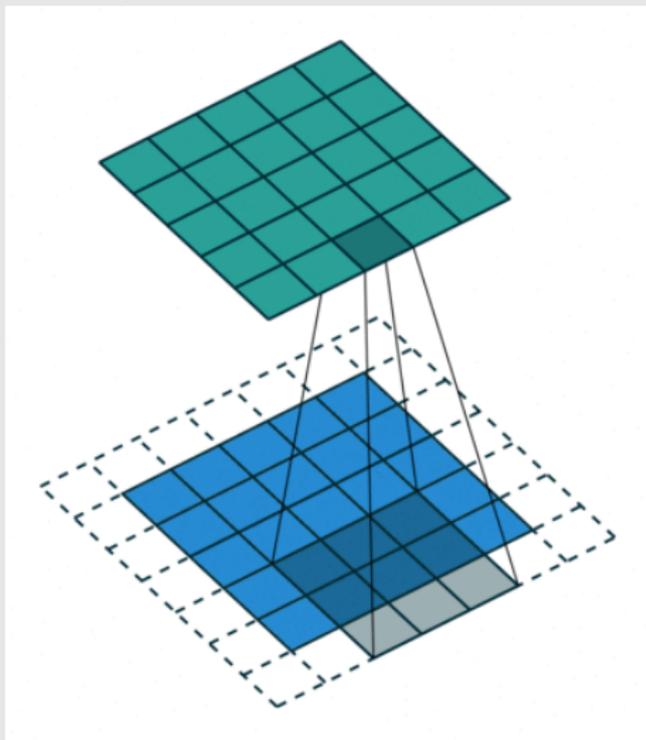


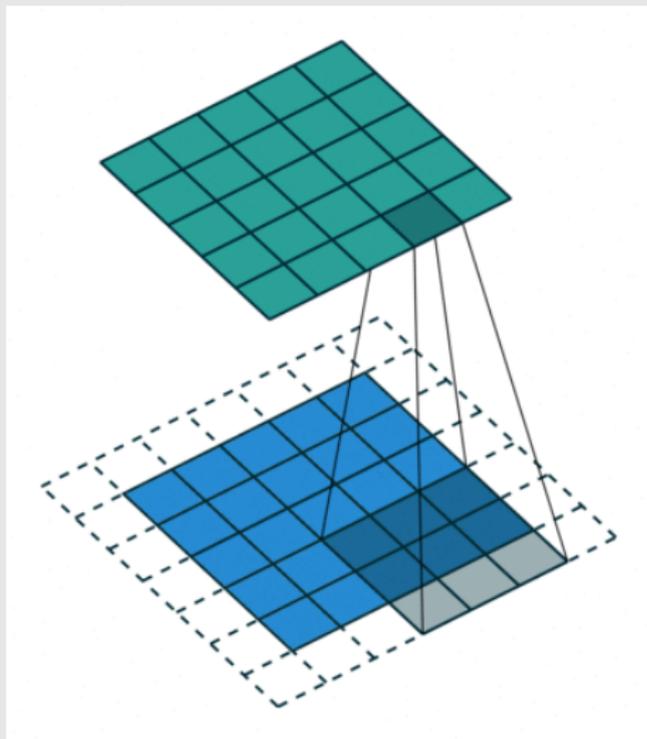


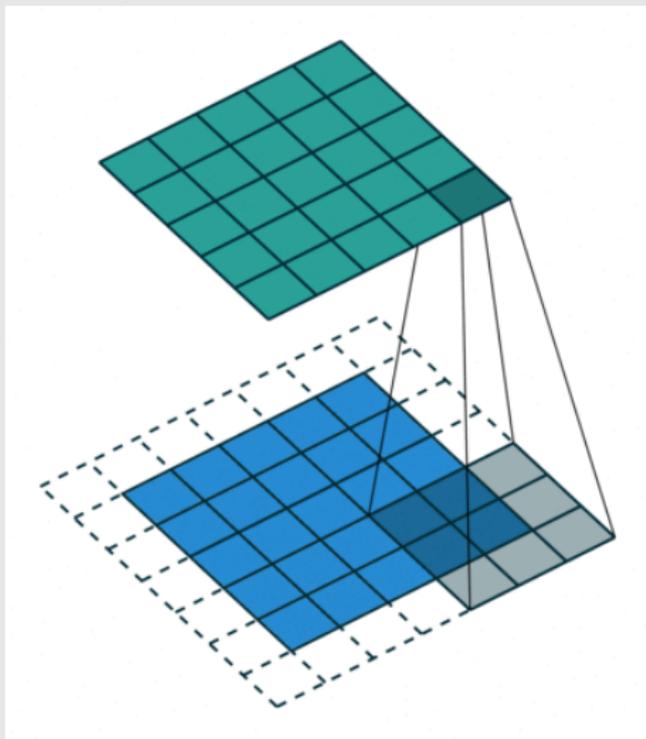












Filtration

- ▶ La taille du filtre est variable.
- ▶ En filtration, on prend des noyaux carrés, de taille impaire. Ça permet donc d'être *centré* sur un pixel.
- ▶ Avec un *padding* de 1, et un glissement de taille un 1, on peut conserver l'image de la même taille. (Comme dans l'animation précédente).

Filtration

Depuis la numérisation de la photo, il est commun d'appliquer des filtres sur des images.

Un filtre est un noyau de taille impaire afin d'être centré sur un pixel et qui permet de modifier l'image d'une certaine manière.

Le filtre suivant:

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

fera une moyenne d'un pixel et de ces 8 voisins le plus proches.

Cela vient *lisser* l'image et réduire l'impact des petits détails. On appelle ces filtres des **flous**.

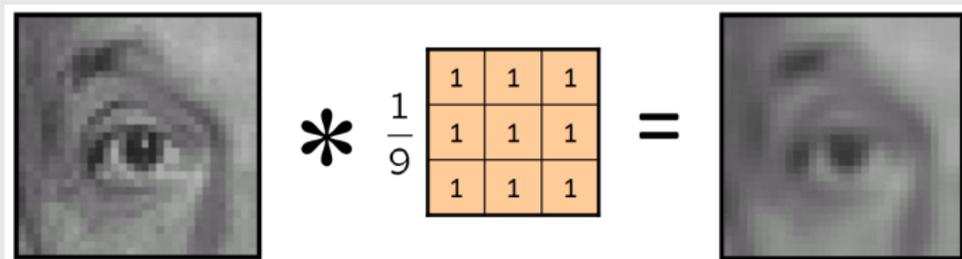


Figure: Lisser l'image corrige les imperfections.

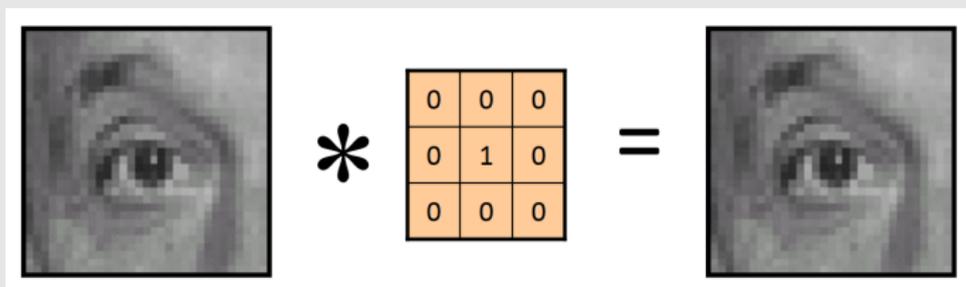


Figure: Filtre identité

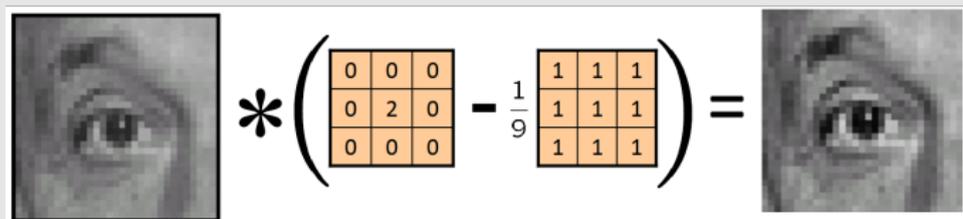

$$\text{Image} * \left(\begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 2 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} - \frac{1}{9} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \right) = \text{Sharpened Image}$$

Figure: On peut soustraire le lissage pour accentuer les détails.
(*Sharpened*)

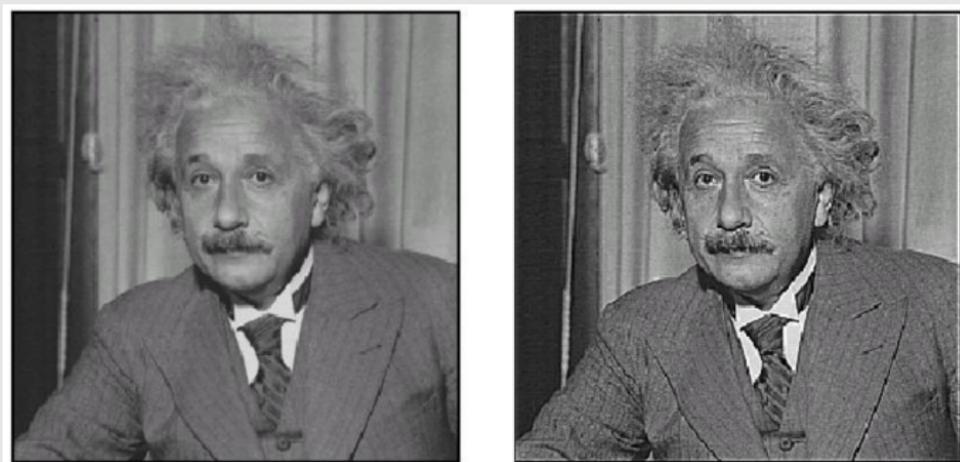


Figure: *Sharpening.*

Détection de contour

Parlons de détection de contour; cela nous servira d'un bel exemple pour contraster des approches fondées sur la représentation par pixel avec ce que nous proposons plus tard.

Ce problème vise donc à identifier la forme séparant les objets dans les images.

Détection de contour

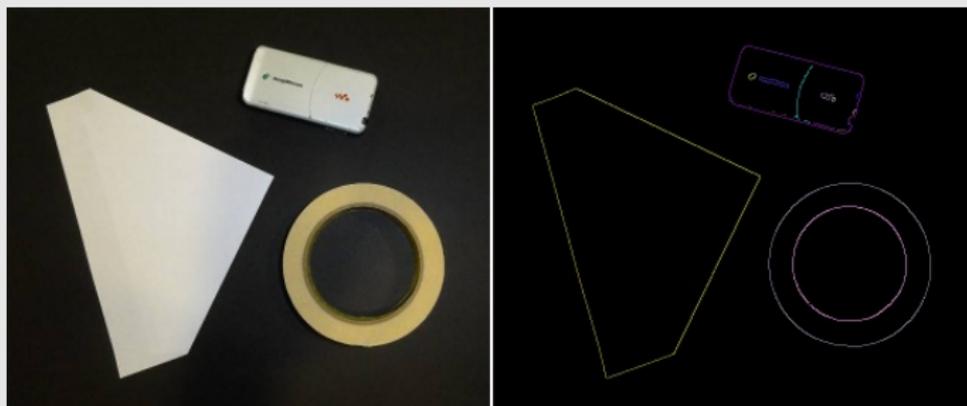


Figure: Détection de contour.

Détection de contour

Une image est une surface 2D que nous observons sur un intervalle discret, une grille de pixel.

Un contour est un changement de couleur drastique.

On veut donc mesurer la variation de $I(i, j)$, la fonction qui retourne la couleur de l'image, en tout point.

Détection de contour

Plus précisément, on veut calculer la variation instantanée de couleur.

En d'autres mots on veut calculer la dérivée sur une image!

Opérateur de Sobel

L'opérateur de Sobel estime le gradient de l'intensité de couleur en chaque pixel.

À cette fin, on emploie la filtration!

Opérateur de Sobel

Les filtres de Sobel sont donc définis comme:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ et}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

et sont en quelques sortes des estimateurs discrets de gradient en longueur (x) et en hauteur (y) de l'image.

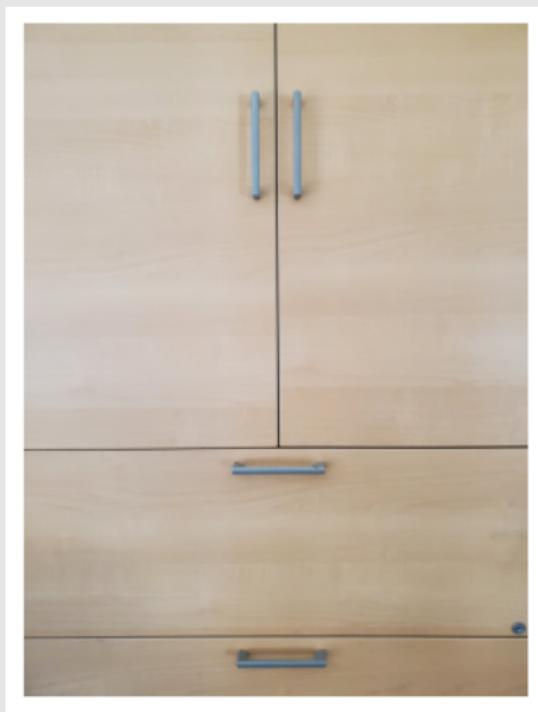


Figure: Mon meuble de rangement, en couleur.



Figure: Mon meuble de rangement, en teinte de gris.

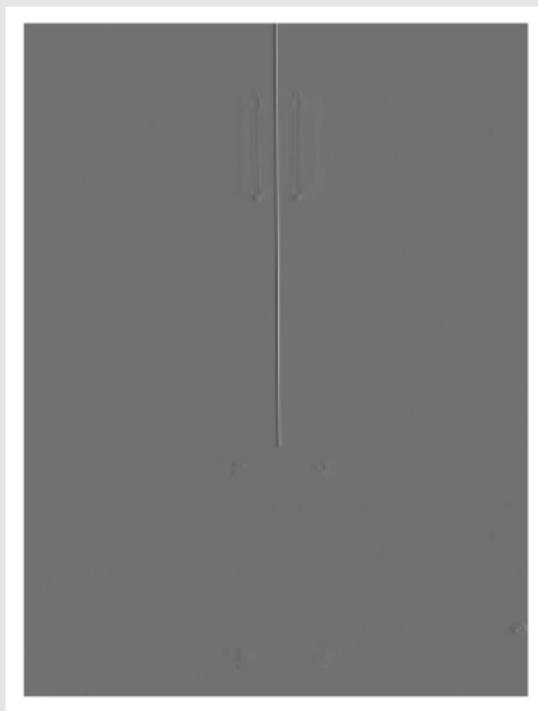


Figure: Après filtration de Sobel (en longueur).

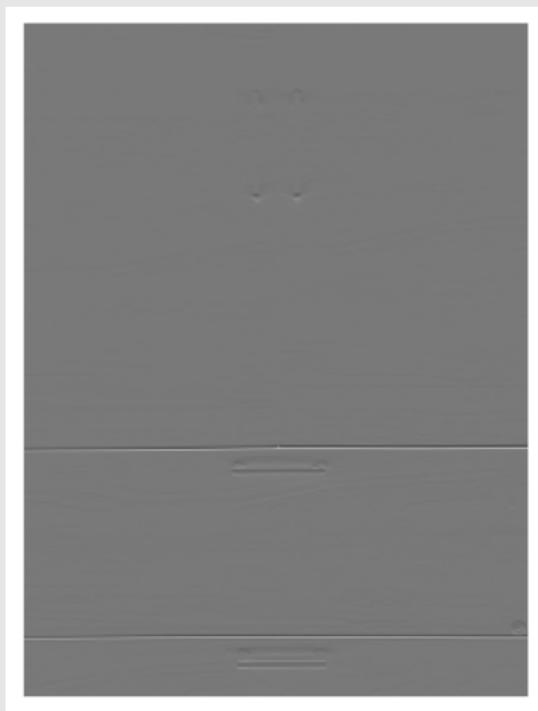


Figure: Après filtration de Sobel (en hauteur).

Méthode Canny

1. Application d'un flou pour diminuer l'impact des artefacts visuel.
2. Calcul de la dérivée sur les pixels (Opérateur de Sobel).
3. Identifier les grandes valeurs de dérivées comme étant des contours.

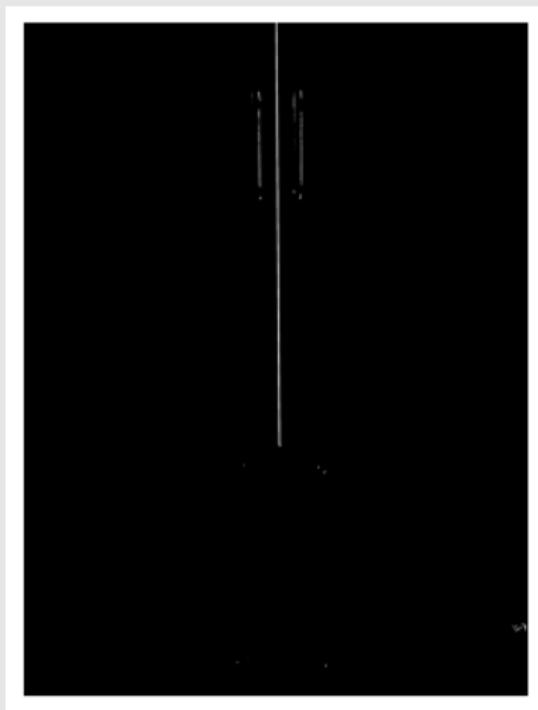


Figure: Après avoir filtré avec Sobel et ne garder que les grandes valeurs (en longueur).

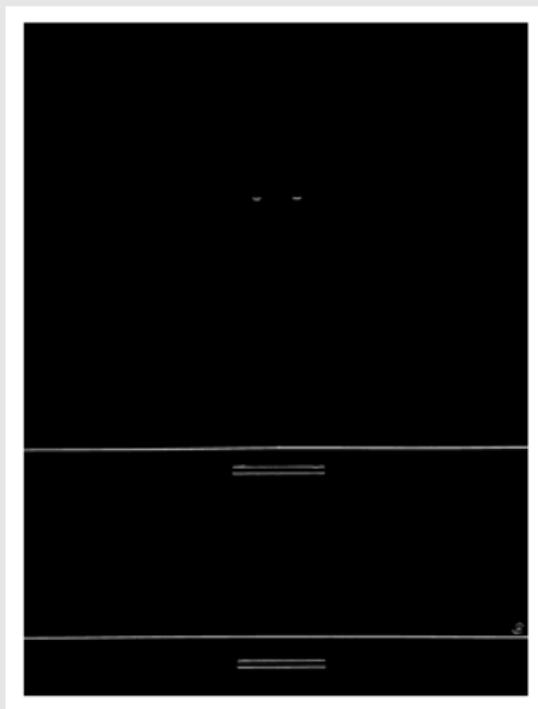


Figure: Après avoir filtré avec Sobel et ne garder que les grandes valeurs (en hauteur).

Détection de contour

Bien qu'il s'agisse d'un problème qui semble relativement simple et que plusieurs solutions furent proposées.

Il n'en demeure pas moins qu'il s'agit d'un problème difficile pour plusieurs raisons.

Plusieurs paramètres doivent être finement ajustés.

Détection de contour

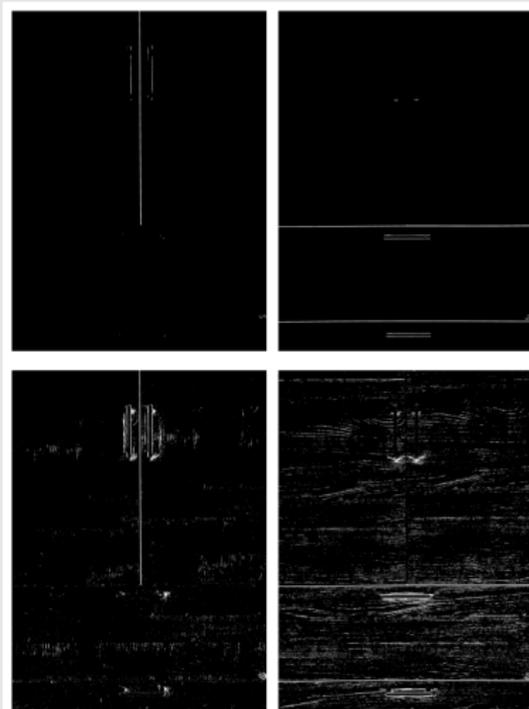
Comment déterminer ce qu'est une grande valeur de dérivé, et s'assurer de former un contour fermé.



Figure: Une image et ses contours en fonction de deux valeurs différentes de sensibilité. (Grandes valeurs de dérivées).

Détection de contour

Différents seuils de grandes dérivées:



Détection de contour sur grille de pixel

L'utilisation des pixels ici cause beaucoup de problèmes.

Les paramètres des outils de détection de contours doivent être ajustés pour chaque image de différentes résolutions.

La même image de différentes résolutions peut émettre des contours significativement différents.

Les formes courbes et lisses ne peuvent être capturées sur une grille.



Figure: Image d'un '5' provenant de la base de données MNIST.

Données fonctionnelles

Données fonctionnelles

L'analyse de données fonctionnelles est un nouveau domaine de la statistique.

On s'intéresse à l'analyse de fonctions $x(t)$, des courbes, ou bien de fonctions de plus haute dimension tel $x(i, j)$ des surfaces.

Données fonctionnelles

En analyse de données fonctionnelles, une observation $x_i(t)$ $t \in T$ est une fonction.

L'espace sur lequel la fonction est défini, T peut être le temps, une surface ou n'importe quelle espace multidimensionnelle.

Un ensemble de données est donc une collection de ce type d'observations: $S = \{x_i(t) | i \in (1, \dots, n)\}$ défini sur le même espace T .

Par contre, il est impossible de collecter et entreposer $x(t) \forall t \in \mathcal{T}$.

Conséquemment, les données sont observées puis collectées de manière discrète.

Pour cette introduction, supposons que \mathcal{T} est un espace de dimension 1, par exemple le temps.

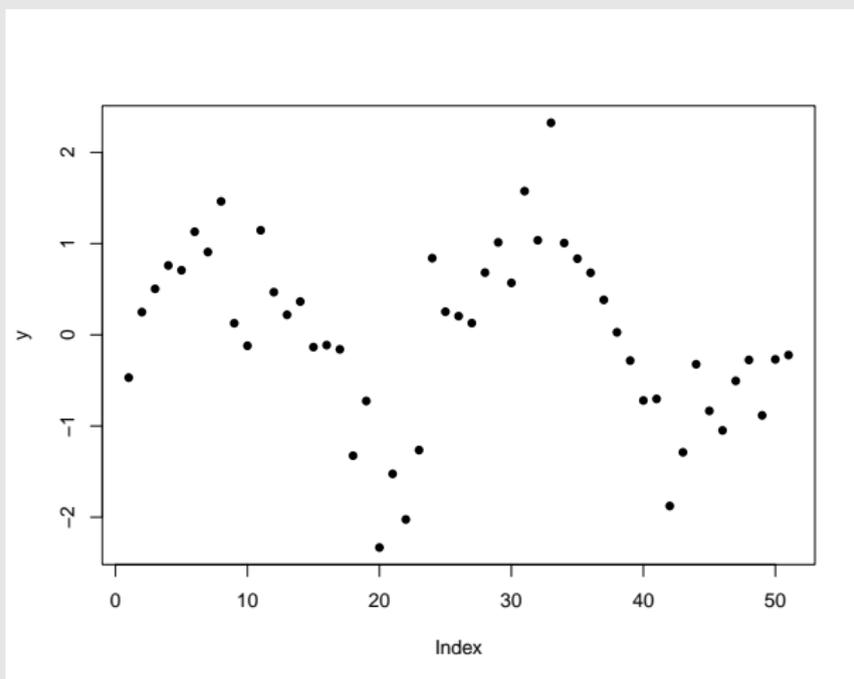


Figure: L'observation d'une variable fonctionnelle.

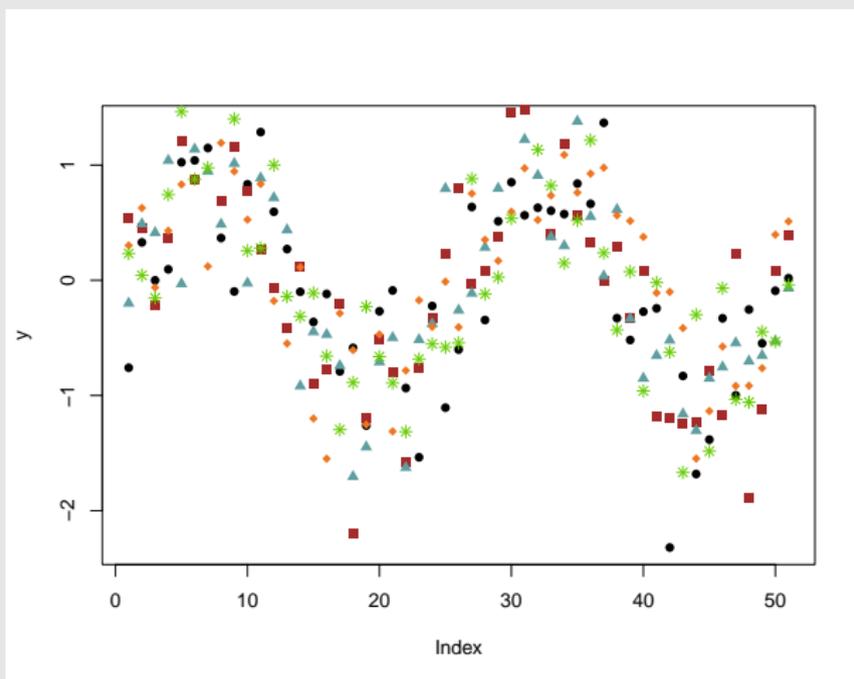


Figure: Un ensemble de données fonctionnelles.

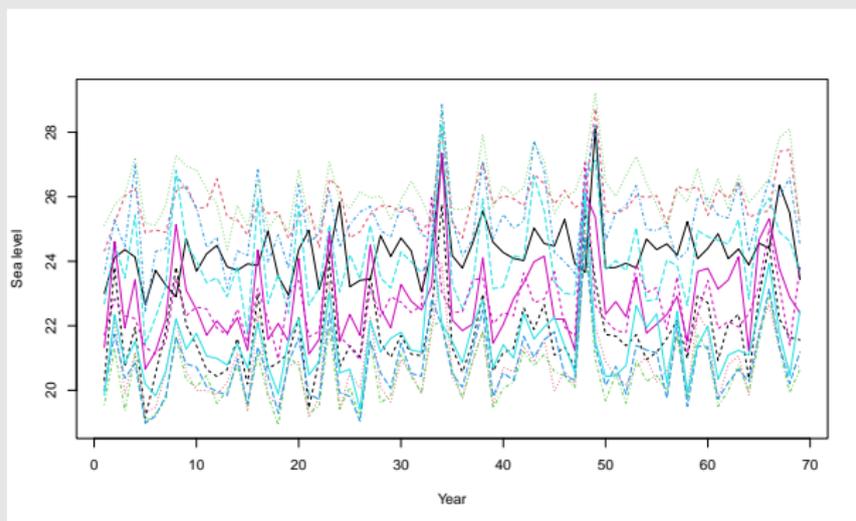


Figure: Température au niveau de la mer sur 70 ans à différentes stations météo.

Exemple:

- ▶ Patient qui souffre de diabète et qui mesure son taux de sucre tous les jours.
- ▶ Température et variable météorologique collectées à travers les années.
- ▶ Signal électrique dans le cerveau collecté par un électroencéphalogramme.

Lissage et représentation fonctionnelle

- ▶ **Nous supposons que les données fonctionnelles observées sont un échantillon (avec bruit) provenant d'un processus lisse sous-jacent.**
- ▶ Donc, afin d'amoindrir l'impact du bruit lors de l'échantillonnage et afin de capturer le réel processus sous-jacent, il est commun de lisser les données fonctionnelles observées.
- ▶ La représentation lisse est par la suite utilisée lors de l'analyse.

Représentation fonctionnelle

- ▶ L'approche standard est d'utiliser une base fonctionnelle; une collection de fonctions de base qui couvre le domaine \mathcal{T} et des coefficients attachés à ces fonctions.
- ▶ La représentation fonctionnelle, lisse et continue est donc paramétrique.
- ▶
$$x(t) = \sum_{k=1}^K c_k B_k(t)$$

Représentation fonctionnelle

- ▶ $x(t) = \sum_{k=1}^K c_k B_k(t)$
- ▶ C'est bien puisqu'on peut évaluer $x(t)$ partout ($t \in [0, T]$).
- ▶ On a seulement besoin *d'apprendre* un petit nombre (discret) de paramètres pour produire cette représentation lisse et continue pour une observation fonctionnelle.
- ▶ On peut souvent le faire facilement en minimisant la fonction d'erreur quadratique suivante: $\sum_j [x(t_j) - \sum_{k=1}^K c_k B_k(t_j)]^2$

B-Splines

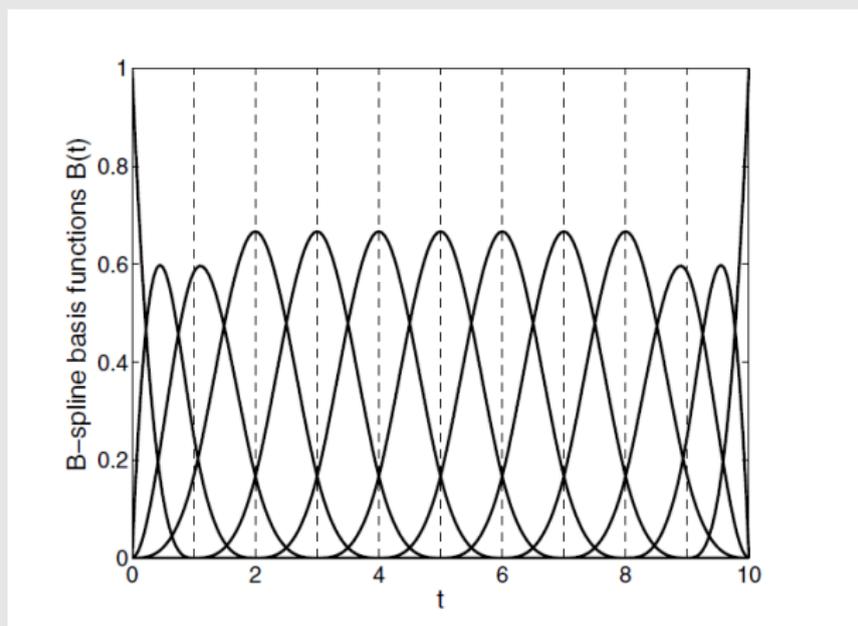


Figure: Les treize fonctions de bases qui forment la base d'ordre quatre sur 10 intervalles (9 noeuds).

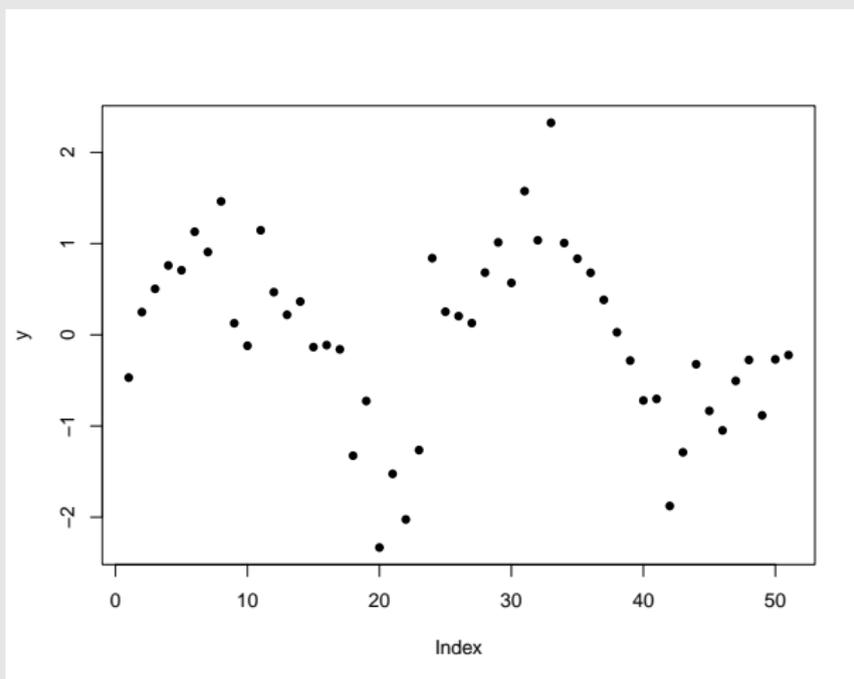


Figure: L'observation d'une variable fonctionnelle.

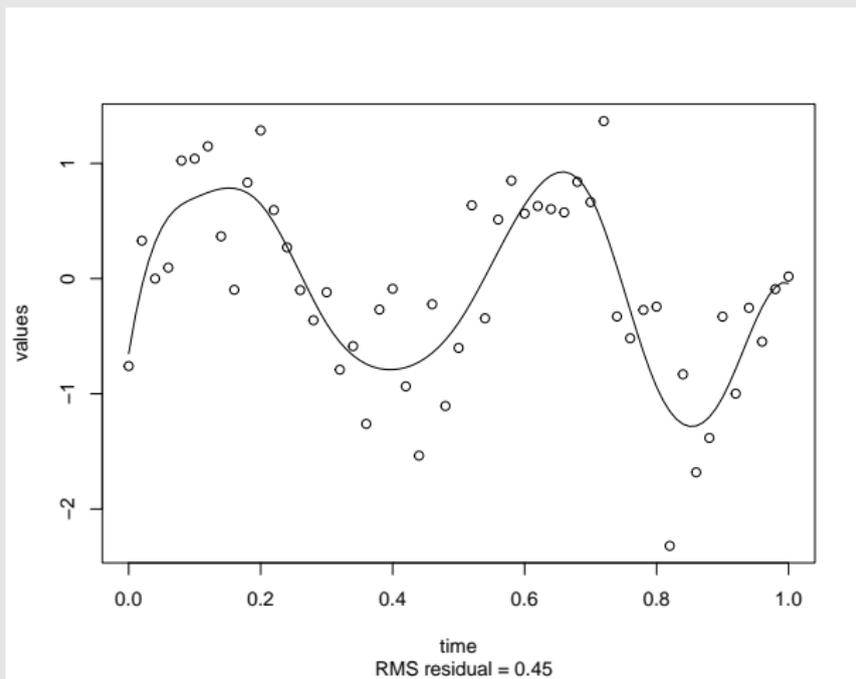


Figure: L'observation d'une variable fonctionnelle et sa représentation lisse.

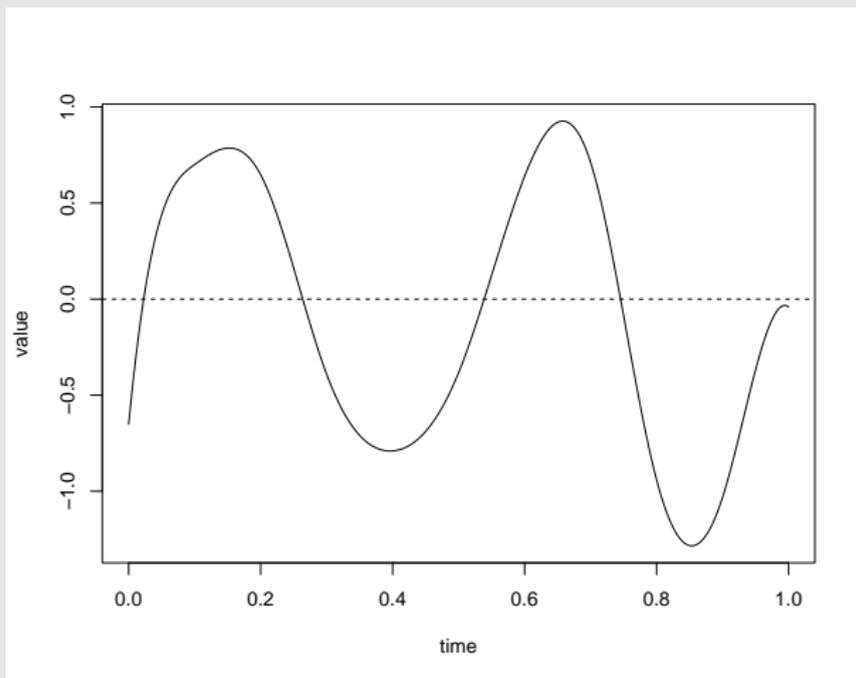


Figure: La représentation fonctionnelle et lisse de l'observation fonctionnelle observée discrètement.

Représentation fonctionnelle

- ▶ On dit simplement que maintenant les données sont représentées par un système B-Splines (ici d'ordre 4 avec 9 noeuds)
- ▶ C'est l'observation lissée qu'on utilise dans nos analyses.
- ▶ Essentiellement les coefficients suivants: $[-0.65115973, 0.53578405, 0.70073762, 0.94566931, -0.59899313, -0.89844612, -0.54772926, 0.79263628, 1.21055785, -1.36245376, -1.40150503, 0.05021092, -0.04074864]$

Représentation fonctionnelle

Le lissage offre plusieurs avantages:

- ▶ Amoindrie la variabilité indésirée.
- ▶ Une réduction de la dimension.
- ▶ Facilite le calcul des dérivées.
- ▶ Permet d'analyser des données collectées irrégulièrement.

Calcul des dérivées

Sous l'hypothèse que le réel processus d'intérêt est lisse et continu; il est raisonnable de croire que les dérivées de ce processus peuvent nous intéresser.

Exemple: on a le poids $w(t)$ d'un patient après une opération, la dérivée est un meilleur indicateur de l'état de santé du patient selon plusieurs études.

Calcul des dérivées

Bien qu'il existe des estimateurs de dérivées discrets, la dérivée est un opérateur usuellement défini pour des fonctions lisses.

Non seulement la représentation fonctionnelle est lisse et donc dérivable, mais cette dérivée est très rapide à calculer.

Calcul des dérivées

Rappel:

$$x(t) = \sum_{k=1}^K c_k B_k(t) \quad (1)$$

$$\Rightarrow x'(t) = \sum_{k=1}^K c_k B'_k(t) \quad (2)$$

Les fonctions de base $B_k(t)$ sont non seulement faciles à dériver, mais leurs dérivées sont connues et *disponibles* dans les logiciels.

Données collectées irrégulièrement

- ▶ On parle essentiellement de données collectées à différents moments pour différents sujets.
- ▶ Les différentes observations i sont collectées à différents moments $t \in [0, T]$
- ▶ Différentes observations peuvent être collectées un nombre différent de fois.
- ▶ Exemple: taux de sucre dans le sang.

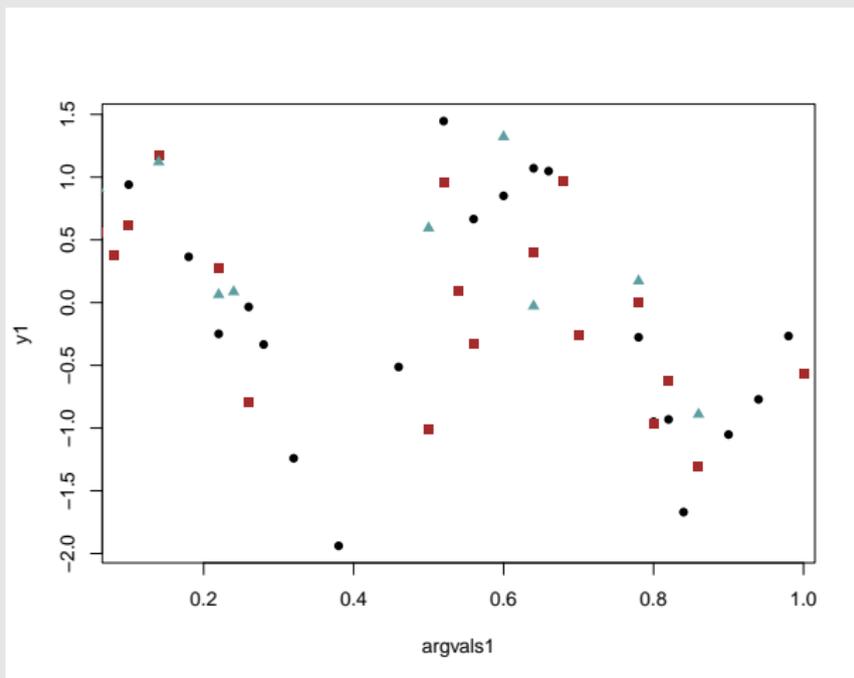


Figure: Exemple d'une base de données fonctionnelles collectées irrégulièrement.

Données collectées irrégulièrement

Si on représente toutes ces observations à l'aide de la même base; elles sont maintenant comparables.

Leurs représentations lisses sont toutes dans le même espace.

Elles sont toutes représentées par le même ensemble de coefficients.

Statistique

Avec une collection de ses courbes, un ensemble de données, on peut on en apprendre sur une population et faire de l'inférence et de la prédiction.

Les patients avec diabète réagissent-ils bien ou mal au traitement?

Quand observe-t-on la plus grande variabilité de température à travers le pays ?

Statistique

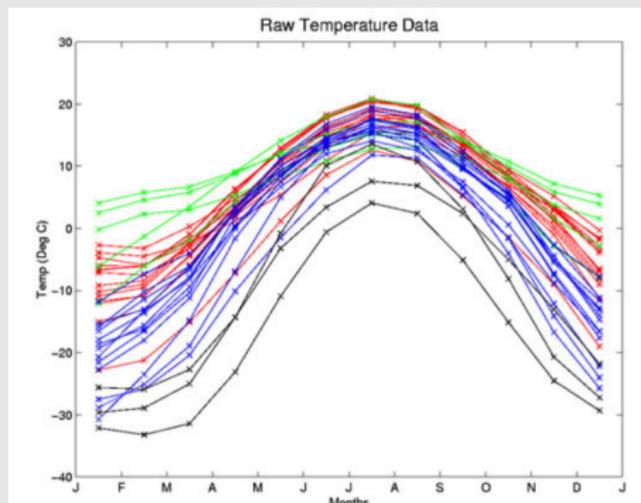


Figure: Échantillon de température sur une année collecté à plusieurs stations météo au pays.

Statistique

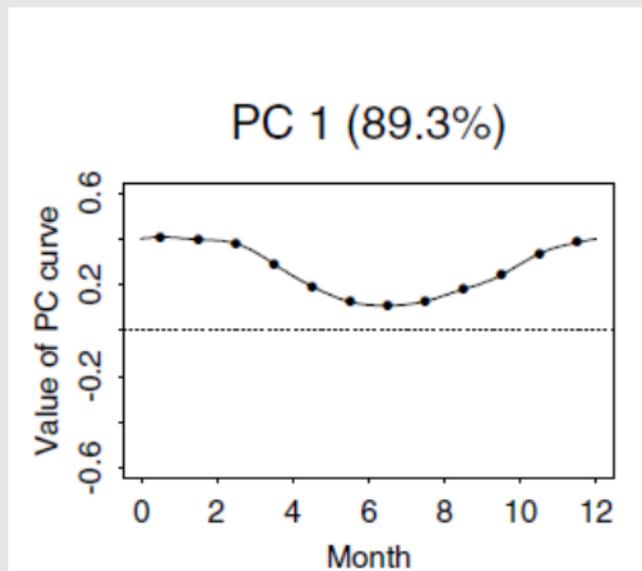


Figure: Première composante principale fonctionnelle associée.

Données fonctionnelles deux-dimensions

Données fonctionnelles deux-dimensions

La majorité de la recherche en données fonctionnelles se consacre aux courbes et fonctions d'un domaine 1-dimension.

Par contre le domaine (et le co-domaine) des fonctions analysées peut-être de plus grande dimension et la plupart des concepts peuvent tout de même s'appliquer.

Mon programme de recherche se concentre sur le développement de nouveaux outils d'analyse d'image, en généralisant certains concepts d'analyse de données fonctionnelles en considérant un domaine 2D.

Données fonctionnelles deux-dimensions

Une image $I(x, y)$ est une observation fonctionnelle 2D, qui varie en fonction de la position (x, y) .

Le domaine de cette fonction est l'étendue de l'image et son co-domaine est $[0, 255]$.

On peut aussi représenter les images comme une surface où l'intensité de coloration varie en fonction de la position (i, j) .

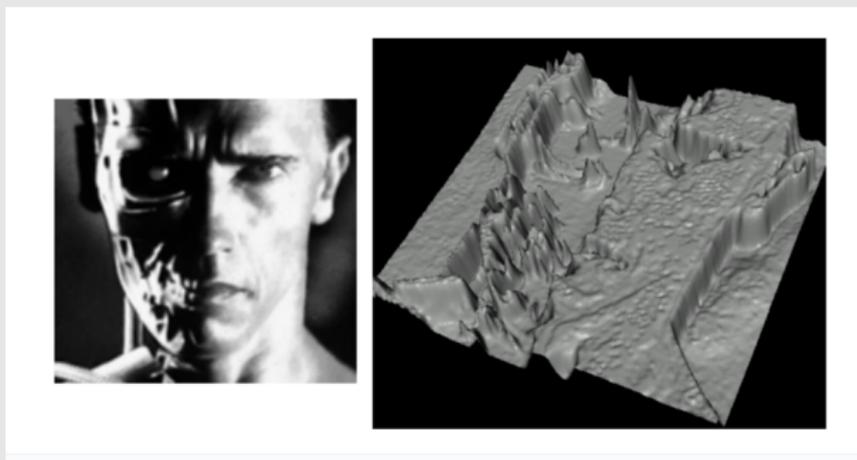


Figure: Sample from a functional data.

La hauteur (z) est donc l'intensité $I_{i,j}$.

Données fonctionnelles deux-dimensions

Parce que les images peuvent être perçues comme une surface 2D, les idées de recherche que je poursuis est l'application de certains concepts d'analyse de données fonctionnelles aux images.

En essence, ce que cela veut dire c'est de s'éloigner du paradigme de l'analyse sur les pixels.

On ne voit plus les images comme une collection de pixels, mais bien comme une surface et l'objectif est donc d'amoindrir certains inconvénients d'une approche par pixel.

Données fonctionnelles deux-dimensions

Le concept de lissage s'applique bien ici.

On perçoit la grille de pixel comme étant une collection d'observations discrètes de la vraie surface sous-jacente.

On utilise cette fois-ci des fonctions Splines bivariées.

et on représente cette surface par: $I(x, y) = \sum_{j=1}^K c_j B_j(x, y)$

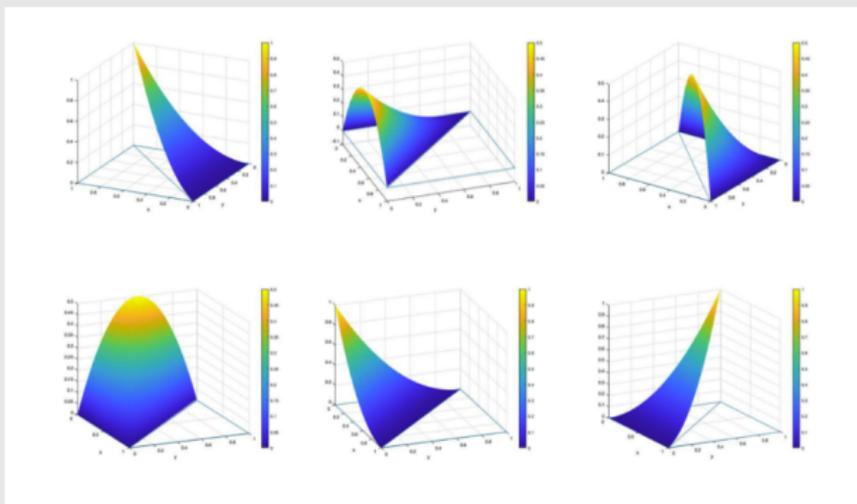


Figure: Exemple de fonctions de base bivariées

Statistique

Récemment, on voit l'apparition des premières études statistiques d'images employant des outils d'analyse de données fonctionnelles. Par exemple, l'analyse en composante principale fonctionnelle fut utilisée pour déterminer les zones de grandes variabilités sur images par résonance magnétique.

Statistique

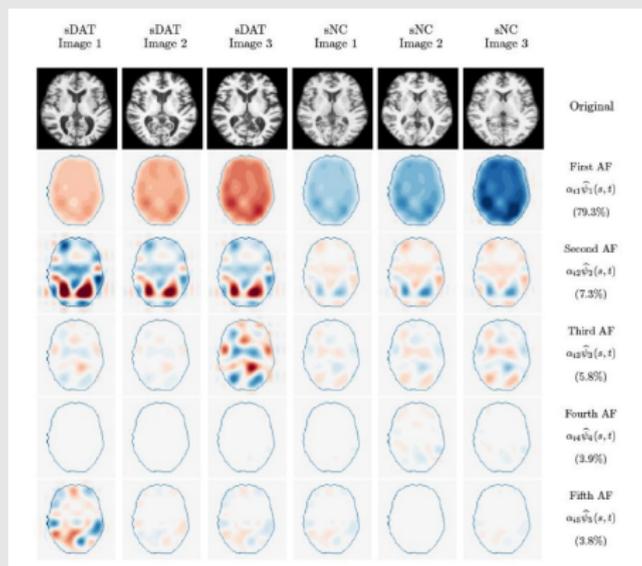


Figure: Les zones de grande variabilité nous aident naturellement à identifier les patients souffrant d'Alzheimer.

Statistique

Une analyse de mammographie fut mise au point pour prédire le cancer du sein en utilisant le mammographie comme variable prédictive.

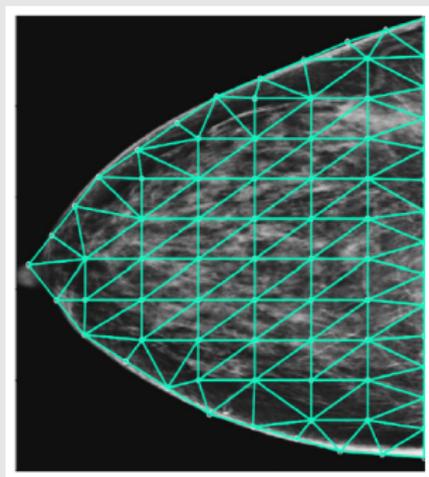


Figure: Mammographie divisée en région, pré-lissage.

Statistique

Une fois un modèle de prédiction mis au point, on peut se demander quelles régions sont importantes pour cette prédiction.

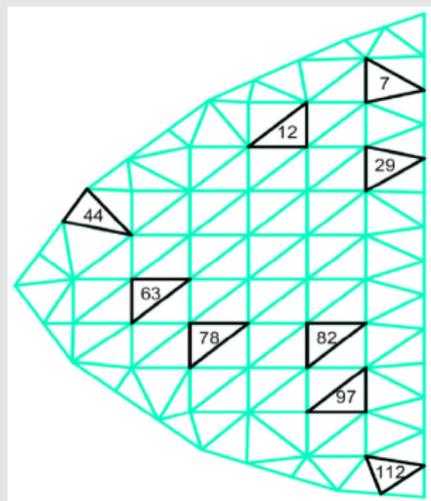


Figure: Identification de région significativement importante dans la prédiction du cancer.

Détection de contour

Dans ce contexte, établir le contour est un défi majeur.

Précédemment nous avons vu la méthode Canny:

1. Application d'un flou pour diminuer l'impact des artefacts visuel (réduction du bruit).
2. Calcul de la dérivée sur les pixels (Opérateur de Sobel).
3. Identification des grandes valeurs de dérivées comme étant des contours.

Détection de contour: projet en cours

Ce que l'on propose:

1. Lissage fonctionnel pour diminuer l'impact des artefacts (réduction du bruit).
2. Calcul de la dérivée directement sur l'image lissé.
3. Identification des points d'inflexions des dérivées comme étant des contours.

Détection de contour: projet en cours

Les avantages de cette approche sont:

- ▶ Il est facile d'identifier les points d'inflexion de la dérivée; le contour de la forme.
- ▶ Le contour appris est lui-même courbe et lisse.
- ▶ En traitant différentes résolutions comme étant des surfaces collectées irrégulièrement; on obtient un système qui fonctionne indépendamment de la résolution des images.

Différentes résolutions

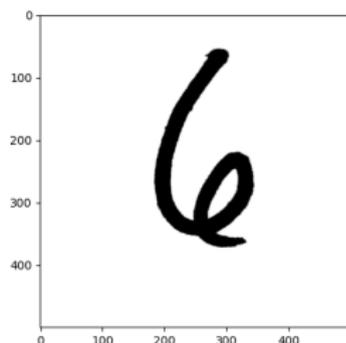
La solution proposée à ce problème est de considérer que différentes résolutions d'image est équivalentes aux données fonctionnelles collectées irrégulièrement.

Pour la même surface, différentes résolutions sont équivalentes à différentes grilles sur lesquelles on observe la surface.

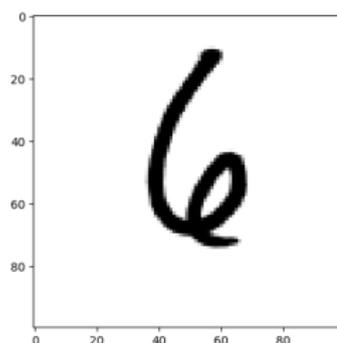
Une fois que toutes ces images sont représentées (lissées) par la même la base, elles sont facilement comparables.

Différentes résolutions

Grille 500x500



Grille 100x100



Formes

Une **forme** est une courbe fermée invariante aux translations, rotations et à l'homothétie.

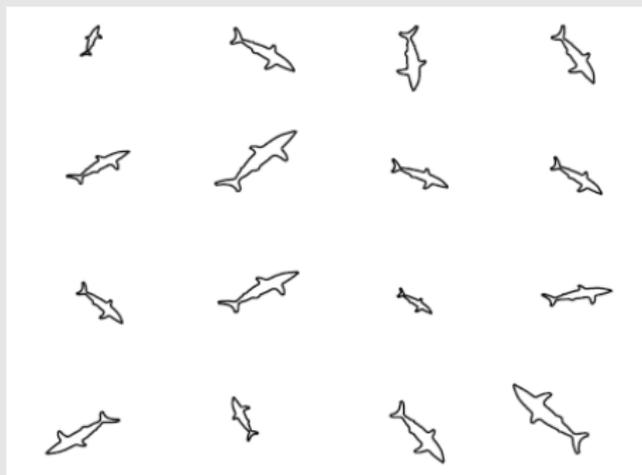


Figure: La même forme après différentes transformations.

Formes

- ▶ Avec des approches fonctionnelles, on veut diviser une image en une collection de formes.
- ▶ On pourrait donc faire de l'apprentissage de formes.
- ▶ On désire capturer la géométrie des objets dans l'image.

Formes: projet en cours

On a des données; des belles images de feuilles, 16 par espèces et une 100aine d'espèces.

On travaille a détecter le contour, le tracer et maintenant former les courbes (coordonnées x et y) pour apprendre une distribution.

Formes: projet en cours

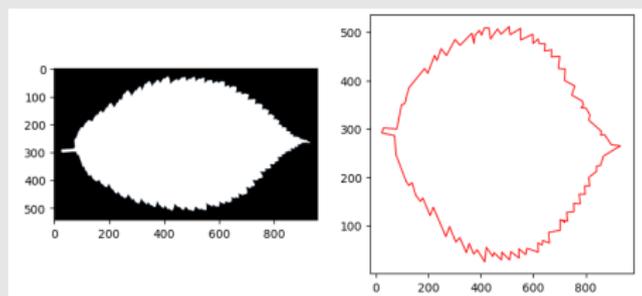


Figure: Le contour d'une feuille.

Formes

Fun-fact, on peut représenter des formes comme des données fonctionnelles 1D.

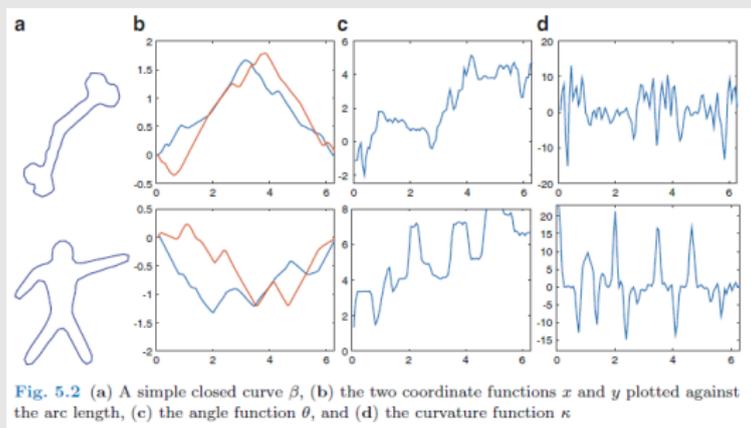


Figure: Représentation fonctionnelle d'une forme (courbe fermée).

Animation ici.

Formes

On peut ainsi appliquer nos connaissances en données fonctionnelles 1D pour estimer une distribution sur ces courbes. On pourrait donc établir des distributions de formes, et même échantillonner de nouvelles formes!

Formes: projet en cours

On espère être en mesure de générer toute sortes de formes de courbes plus tard cet été!

Conclusion

- ▶ L'analyse d'images moderne est fondée sur la représentation en matrice de pixel et la filtration est l'outil de base.
- ▶ L'analyse de données fonctionnelles nous permet d'approximer des processus observés discrètement par des courbes lisses.
- ▶ On veut combiner les deux pour analyser les images comme une surface lisse par bloc.
- ▶ Il faut garder en tête qu'il s'agit d'objets 3D capturés en 2D.

Ramsay, J., Silverman, B.W.(2013). Functional Data Analysis. Switzerland: Springer New York.

Jiang, S., Cao, J., Colditz, G. A. (2023). Identifying regions of interest in mammogram images. Statistical methods in medical research, 32(5), 895-903.

Shi, H., Yang, Y., Wang, L., Ma, D., Beg, M. F., Pei, J., Cao, J. (2022). Two-dimensional functional principal component analysis for image feature extraction. Journal of Computational and Graphical Statistics, 31(4), 1127-1140.

Srivastava, A., Klassen, E. P. (2016). Functional and shape data analysis (Vol. 1, pp. 43-44). New York: Springer.

Forme du contour

Entres autres, comme la grille est plan cartésien; il est difficile de capturer des formes courbes qui ne sont pas alignés avec la grille.

À vrai dire, sans lissage post-traitement, les contours seront détectés sur la grille et donc ne seront jamais lisse et courbe.

Forme du contour

Il est donc significativement plus difficile de bien identifier les formes et les estimer de manière paramétrique.

De la même manière, la forme estimée dépend aussi de son alignement original.

C'est-à-dire que la même forme mène à différents contours suite à des opérations simples telles que la translation et la rotation.

Résolution

Une autre problème de travailler et manipuler les images sur les pixels est que cela rend les modèles dépendant de la résolution.

La résolution étant la densité de la grille discrète sur laquelle nous observons la surface qu'est une image.

Résolution

Non seulement différentes résolutions produiront différentes formes.

En plus, les opérations de lissage par filtres et de détection de contour sont très sensible à la résolution.

Les paramètres devront constamment être ré-ajusté.

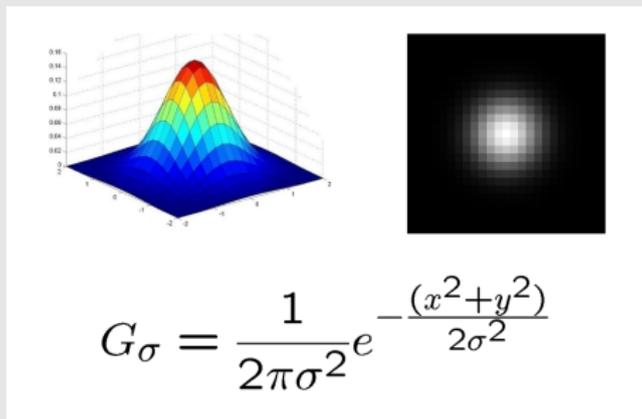


Figure: Sample from a functional data.

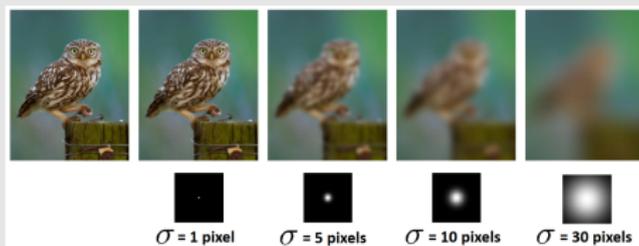


Figure: Sample from a functional data.



Figure: Sample from a functional data.

Problèmes et objectifs

De grands progrès ont été faits en analyse d'images et les résultats. C'est un domaine en expansion constante; en gros ça va bien et le milieu survivrait sans moi.

Par contre, on constate que de travailler directement sur les pixels cause quelque ennuis.

La résolution de l'image et des objets s'y trouvant peut causer certains problèmes.

Changement de résolution

La résolution de l'image et des objets s'y trouvant peut causer certains problèmes.

Comme les opérateurs et fonctions sont définis directement sur les pixels, changer la résolution d'une image ou d'un objet nous force à changer la fonction ou revoir comment l'opérateur se comporte.

Exemple:

Les modèles de prédiction sont des fonctions f prenant en entrée un image, donc une matrice (l, h) et retourne une réponse y disons un diagnostic: $f : I_{(l,h)} \rightarrow [0, 1]$.

Si on apprend une fonction $f : I_{(l,h)} \rightarrow [0, 1]$ et que nous avons maintenant des images d'une nouvelle résolution (l', h') , on ne peut plus utiliser f .

De voir les images comme une collection de pixel et nous on ensemble d'objet géométrique cause aussi toute sorte de problèmes chez les modèles génératifs.

Il est commun de voir beaucoup d'image d'humain à plus de 5 doigts et plus de trois jambes etc...