

# Journée HPC: Apprentissage de métrique appliquée à la Vision par Ordinateur

Greyc, Université de Caen, Ensicaen, CNRS

Alexis Lechervy



4 juin 2018



# Sommaire

- 1 Introduction
  - L'apprentissage automatique
  - Les réseaux de neurone
  - L'apprentissage de métrique
- 2 TS-Net
- 3 Conclusion

# L'apprentissage automatique

## Apprendre ? Qu'est ce que c'est ?

- **Apprendre, c'est s'adapter** à des situations **nouvelles et inconnues** en prenant en compte l'expérience passée.
- Apprendre est une **propriété humaine essentielle**.
- Apprendre signifie **s'améliorer afin d'être meilleur**.
- A ne pas confondre avec la récitation par-coeur. N'importe quel ordinateur peut "apprendre" par-coeur, la difficulté est de généraliser à des situations nouvelles et inconnues.

## Mais pourquoi apprendre à un ordinateur ?

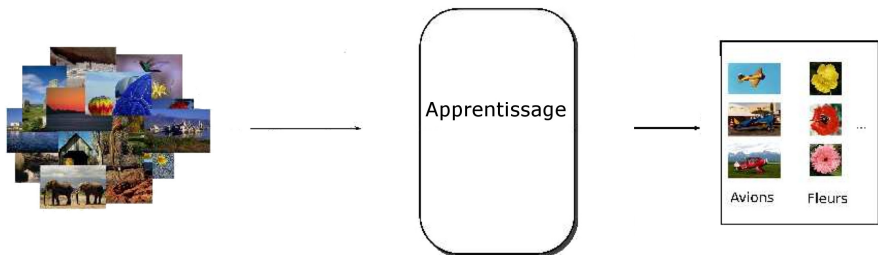
- Pouvoir gérer une quantité de données très importante de manière automatique ou de manière temps réels.
- Pouvoir effectuer une action dans un contexte non prévu préalablement sans l'intervention d'un humain.
- Pouvoir prévoir des comportements ou des évolutions pour aider un humain dans sa prise de décision.

# L'apprentissage supervisé

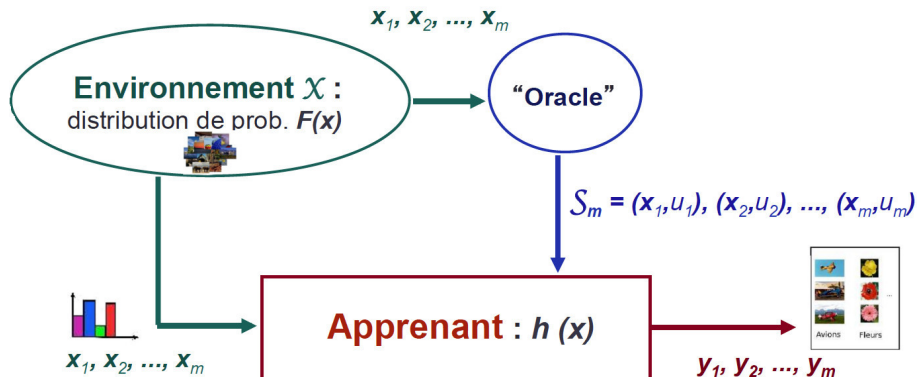
## Principes

- On détermine automatiquement une règle à partir de données d'apprentissage annotées par un expert,
- Un expert a défini un ensemble de couples (donnée,label),
- Il y a un a priori sur les données,
- Les données entrées sont des couples (données collectées, observations).

## Exemple : Catégorisation d'image



# Déroulement classique de l'apprentissage supervisé



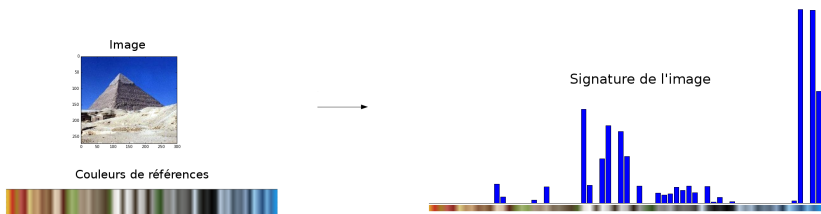
**Induction :** Proposer des **lois générales** à partir de l'observation de **cas particuliers**.

# Étapes classiques d'un apprentissage

## Étapes

- 1 Calcul d'un vecteur de caractéristique à partir des données. Généralement en utilisant un algorithme conçu en fonction de la nature des données.
- 2 Apprentissage de la tâche cible à partir des vecteurs précédents et d'un a priori. Généralement, on cherche à produire une fonction paramétrique minimisant un certain critère d'erreur.

## Exemple de vecteur de caractéristique

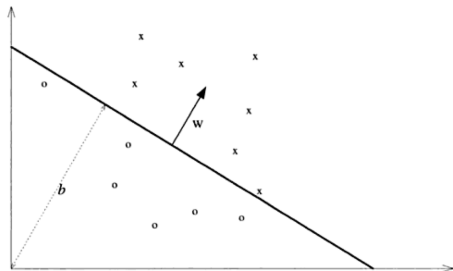


# Étapes classiques d'un apprentissage

## Étapes

- 1 Calcul d'un vecteur de caractéristique à partir des données. Généralement en utilisant un algorithme conçu en fonction de la nature des données.
- 2 Apprentissage de la tâche cible à partir des vecteurs précédents et d'un a priori. Généralement, on cherche à produire **une fonction paramétrique** minimisant un certain critère d'erreur.

## Exemple de fonction paramétrique



$$f_w(x) = \begin{cases} 1 & \text{si } \langle w, x \rangle + b > 0 \\ -1 & \text{sinon} \end{cases}$$

# Étapes classiques d'un apprentissage

## Étapes

- 1 Calcul d'un vecteur de caractéristique à partir des données. Généralement en utilisant un algorithme conçu en fonction de la nature des données.
- 2 Apprentissage de la tâche cible à partir des vecteurs précédents et d'un a priori. Généralement, on cherche à produire une fonction paramétrique minimisant un certain **critère d'erreur**.

## Exemple de fonction de cout

On peut chercher à maximiser la probabilité d'être bien classé en utilisant par exemple la cross-entropie.



# Faire du bio-inspirée ?

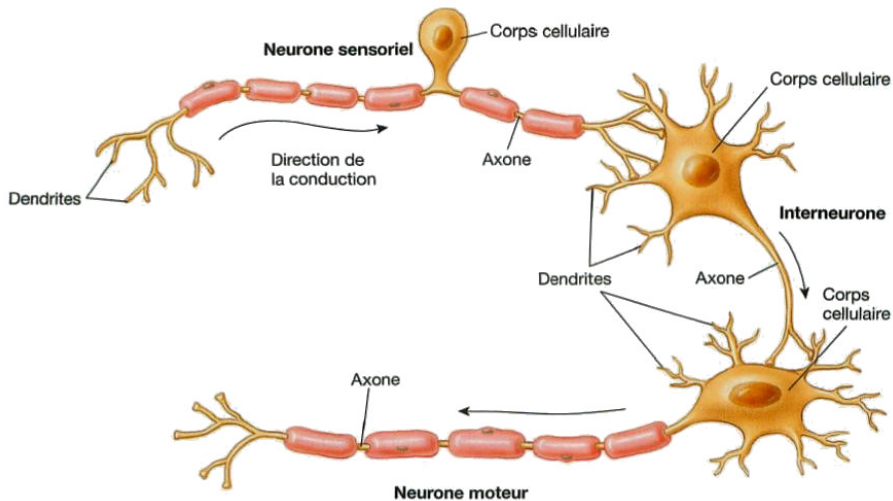
## But

Le cerveau humain excelle dans l'apprentissage de nouvelle situation. Pourquoi ne pas s'en inspirer pour élaborer une méthode d'apprentissage ?

## Les avantages d'un cerveau naturel

- Flexible, peut s'adapter à de nouvelles données .
- Robuste et tolérant aux erreurs dans la base d'apprentissage.
- Peu fonctionner avec des données incomplètes ou en partie fausse.
- A des grandes capacités d'apprentissage.
- Est rapide et utilise un système massivement parallèle.
- Repose sur des éléments dont le fonctionnement est relativement simple et facilement reproductible.

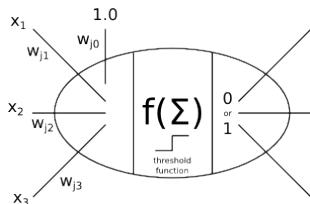
# Du récepteur sensoriel à l'action motrice



# Du neurone réel au neurone formel

## Le neurone formel

- Un neurone possède des entrées  $x_1, \dots, x_d$ .
- Le signal de chaque entrée est pondéré par un "coefficient synaptique"  $w_1, \dots, w_d$ .
- La sortie dépend des entrées et des poids.
- La sortie est un signal d'activation ou non du neurone (par exemple 0 ou 1).



## Apprentissage pour un neurone

L'apprentissage consiste à mettre à jours les poids  $w_i$  pour minimiser un certain critère d'erreur comparant les sorties du réseau et les valeurs attendues.

# Les réseaux de neurones

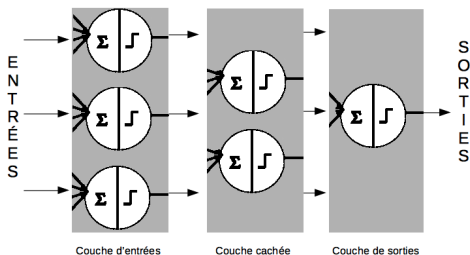
## Définition

Un réseau de neurones est un enchaînement de neurone consistant à appliquer une somme pondérée de ses entrées suivis d'une fonction non linéaire. Il est représenté graphiquement par un ensemble de neurones connectés entre eux, l'information circulant des entrées vers les sorties sans retour en arrière possible.

On distinguera trois types de couches neuronales :

- 1 la couche d'entrée,
- 2 les couches cachées,
- 3 la couche de sortie.

## Exemple de réseau de neurones non bouclé



# Sommaire

## 1 Introduction

## 2 TS-Net

- Contexte
- La méthode
- Le protocole expérimental
- Les ressources utilisées
- Les résultats

## 3 Conclusion

# L'apprentissage de fonction de similarité

## Objectif

Notre objectif est d'apprendre une fonction de similarité, permettant de comparer deux images est de reconnaître si elles sont ou non similaire selon un critère sémantique.

## Exemples d'application

- Reconnaissance faciale



Janica Kostelic, 1



Janica Kostelic, 2



Jim O'Brien, 1



Jim O'Brien, 3

- Reconnaissance photo/dessin

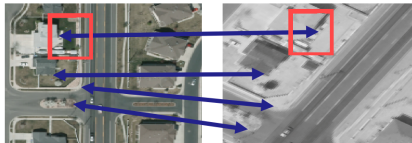


Alexis Lechervy (UNICAEN)

- Appariement de zone de deux images



Image from (Rapantzikos et al. [2010])



# Apprentissage sur des données multi-modales

## Objectif

Notre objectif est d'apprendre sur des données de natures différentes.

## Exemples

- Photo/Dessin

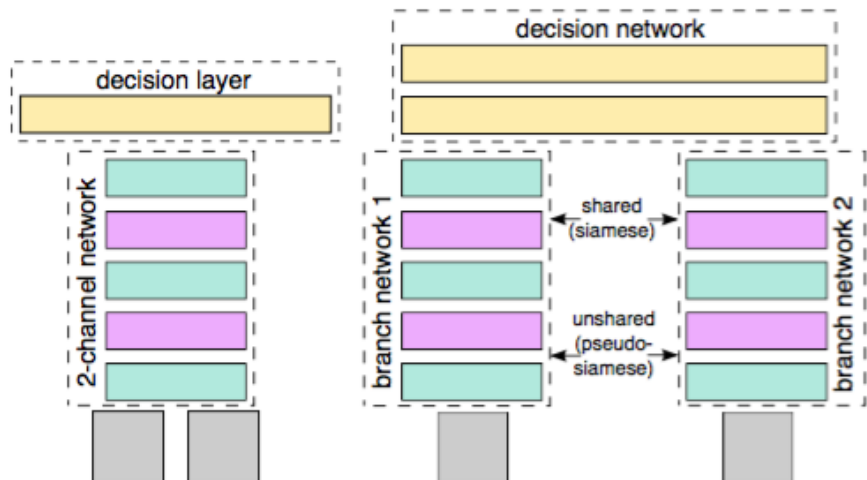


- Visible/Infra-rouge



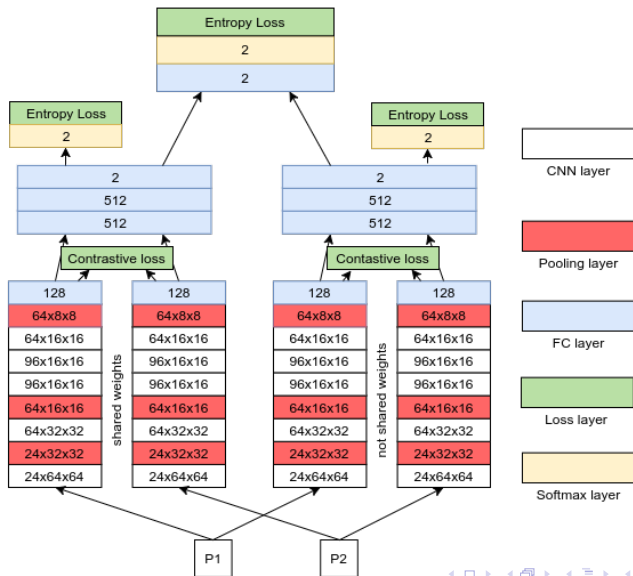
# L'existant

Réseaux multi-canaux, siamois et pseudo-siamois



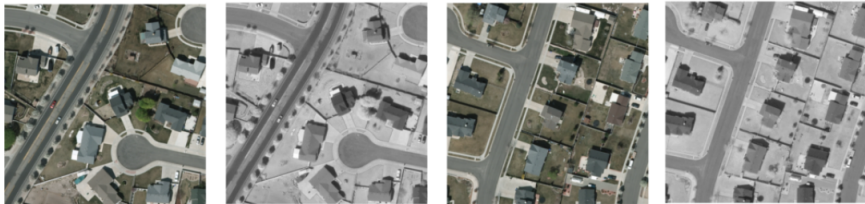


# Architecture proposée : TS-Net (a three-stream network)



# Les bases utilisées : VEDAI

**VEDAI**<sup>1</sup> [RJ16]: is a dataset for Vehicle Detection in Aerial Imagery, provided as a tool to benchmark automatic target recognition algorithms in unconstrained environments.



## Les bases utilisées : Nir Scene

**RGB-NIR Scene**<sup>2</sup>: consists of 477 images in 9 categories captured in RGB and Near-infrared (NIR). The images were captured using separate exposures from modified SLR cameras, using visible and NIR filters. The scene categories are: country, field, forest, indoor, mountain, old building, street, urban, water



## Les bases utilisées : CUHK

**CUHK Face**<sup>3</sup>: is for research on face sketch synthesis and face sketch recognition. In total, there are  $\approx 300$  faces in total. For each face, there is a sketch drawn by an artist based on a photo taken in a frontal pose, under normal lighting condition, and with a neutral expression.



# Configuration technique et temps de calcul

## Configuration matérielle du calculateur Myria

- Cartes graphiques P100 ou K80,
- Nombre de coeur CPU utilisé 6 à 8,
- Mémoire 5Go.

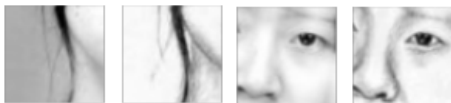
## Configuration logiciel du calculateur Myria

- Python 3.6.1,
- Tensorflow 1.4.0,
- Cuda 9.0, cuDNN 7.0

## Temps de calcul pour l'apprentissage

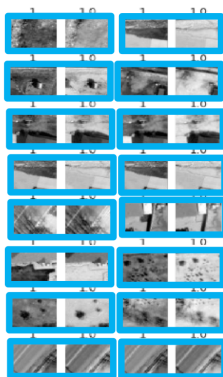
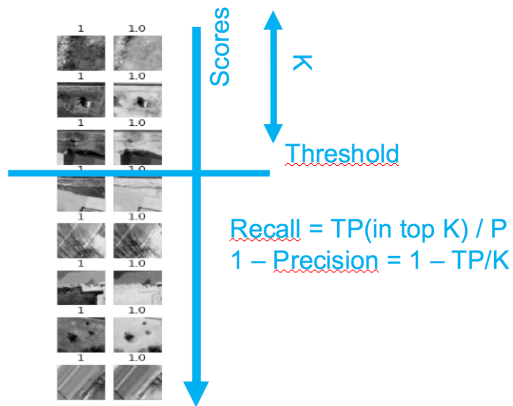
- Vedai  $\sim 8h$  (448k pairs pour l'apprentissage, 64k pour l'évaluation et 128k pour le test),
- CUHK  $\sim 3h$ , (113k pairs pour l'apprentissage, 16k pour l'évaluation et 32k pour le test)
- RGB-NIR Scene  $\sim 8h$  (427k pairs pour l'apprentissage, 122k pour l'évaluation et 128k pour le test).

# Résultats qualitatifs : CUHK



# Critère d'évaluation des performances

- Measure of performance: 1-Precision @95% of recall



True positive

# Résultats quantitatifs

Table – 95%ErrRate on the 3 datasets, for Siamese network alone (S), Pseudo-Siamese network alone (PS), TS-NET, without/with the additional contrastive loss (C).

Dataset	Vedai	CUHK	NIR Scene
S	$1.16 \pm 0.07$	$5.07 \pm 0.46$	$14.35 \pm 0.20$
PS	$1.50 \pm 0.08$	$5.56 \pm 0.36$	$16.05 \pm 0.30$
TS-Net	$0.52 \pm 0.07$	$3.58 \pm 0.14$	$12.40 \pm 0.34$
S+C	$0.84 \pm 0.05$	$3.38 \pm 0.20$	$13.17 \pm 0.86$
PS+C	$1.37 \pm 0.08$	$3.70 \pm 0.14$	$15.60 \pm 0.28$
TS-Net+C	<b><math>0.45 \pm 0.05</math></b>	<b><math>2.77 \pm 0.07</math></b>	<b><math>11.86 \pm 0.27</math></b>

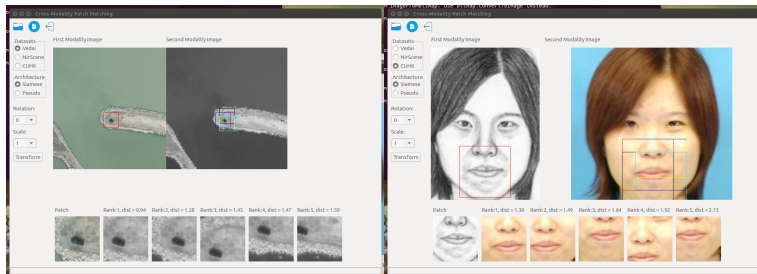


# Démonstrateur

## Principe

On apprend sur les serveurs du Criann puis on test sur une machine de bureau.

## Le démonstrateur



# Sommaire

- 1 Introduction
- 2 TS-Net
- 3 Conclusion

# Conclusion

## Résumé

- Développement d'une nouvelle architecture de **réseau de neurone** pour l'**apprentissage de métrique** dans un contexte **multi-modal**.
- Architecture en **trois sous-réseaux** complémentaire.
- Utilisation de **plusieurs fonctions de coût** à différent niveau.
- **Gain de performance** notable par rapport aux architectures classiques.

## Pour aller plus loin

TS-NET : Combining modality specific and common features for multimodal patch matching.

Sovann En, Alexis Lechervy et Frédéric Jurie. IEEE International Conference on Image Processing (ICIP) 2018.