



**Concours externe Inria 2017**

**Arrêté du 9 juin 2017**

**Poste « DT1 Ingénieur de développement et d'expérimentation – Science des données  
(h/f)»**

**Accès au corps des « Ingénieurs de Recherche»**

**Epreuve du 26 septembre 2017**

**Note sur 20 – Coefficient 3 – Durée 3 heures**

\*\*\*\*\*

*La notation prendra en compte la qualité des réponses, mais aussi la rédaction, la présentation, le style et l'orthographe.*

*Veillez respecter l'anonymat dans les réponses.*

*Ne pas omettre de noter votre numéro d'ordre sur les feuilles intercalaires.*

\*\*\*\*\*

## Exercice 1 (médiation scientifique – 3 points)

Le centre Inria de Paris va recevoir des classes de collèges dans le cadre de la fête de la science (âge des collégiens : entre 14 et 15 ans). L'équipe de recherche Willow qui travaille autour de reconnaissance visuelle d'objets et de scènes est sollicitée pour présenter ses travaux à ces collégiens. Vous êtes désigné pour faire cette présentation.

Le format est de 30 mn par groupe de 15 collégiens. On vous demande de mettre le focus sur les techniques de réseaux de neurones, et deep learning, et d'illustrer avec des résultats de recherche de l'équipe Willow.

### Question (longueur du plan détaillé : max 20 lignes, nombre de slides max : 8)

Proposez un plan détaillé pour cette présentation, en précisant le timing associé, et l'illustration que vous allez utiliser, et de quelle manière vous allez l'exploiter.

De plus, rédigez les points principaux de votre plan sous forme de « slides ».

Note : Vous trouverez en Annexe A des informations complémentaires concernant les axes de recherche de l'équipe Willow, ainsi que certains de ses résultats de recherche.

## Exercice 2 (gestion de projet – 4 points)

Anthony B., doctorant de l'équipe Inria ABC en BioInformatique, a développé un outil lui permettant de visualiser rapidement des jeux de données disponibles dans le cadre d'un projet collaboratif en Machine Learning (ML), et de faire tout un ensemble d'opérations dessus (e.g., nettoyage et d'enrichissement, filtrage, transcodage, visualisation de corrélations et des distributions,.....). Les jeux de données sont principalement médicaux et proviennent de différents partenaires.

Marc G., responsable de l'équipe ABC souhaite continuer le développement de cet outil et fait donc une demande de soutien (voir Annexe B) pour disposer d'un budget pour recruter un ingénieur pendant un an et demi (18 mois).

### Question 1 (2 points, 1/2 page maximum)

Relevez les points forts et points faibles de cette demande, évaluez la validité du plan de travail proposé, et donnez un avis détaillé quant à la pertinence de cette demande.

### Question 2 (2 points, 1/2 page maximum)

Rédigez **EN ANGLAIS** un retour au porteur du projet en fonction de ce que vous avez évalué (points forts, points d'améliorations, points à revoir, ...).

## Exercice 3 (mathématiques – 5 points)

### Question 1 (0,5 point)

$$\text{Soit } J = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Calculez  $J^k$ , pour tout  $k = 0, 1, 2, \dots$

**Question 2.a (1 point)**

Soit  $y \in \mathbb{R}^n$  et  $X \in \mathbb{R}^{n \times p}$ . On note  $\hat{\beta} \in \mathbb{R}^p$  une solution du problème d'optimisation (estimateur des moindres carrés) suivant :  $\min_{\beta \in \mathbb{R}^p} |y - X\beta|$

Sous l'hypothèse que  $X^T X$  est inversible, donnez une formule sous forme analytique (closed-form formula) pour  $\hat{\beta}$ .

Dans le cas où la matrice  $X^T X$  n'est pas inversible, que proposez-vous de faire pour obtenir un estimateur pertinent ?

**Question 2.b (1 point)**

Proposez au moins deux algorithmes calculant les solutions du problème précédent, et discuter de leur forces et faiblesses.

**Question 2.c (1 point)**

Reprenez la question 2.b mais cette fois avec la contrainte que les coefficients  $\beta$  sont positifs ou nuls.

**Question 3.a (0,5 point)**

On rappelle l'expression de la densité d'un variable aléatoire Gaussienne univariée de moyenne  $\mu$  et de variance  $\sigma^2$  :  $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-(x-\mu)^2}{2\sigma^2}$ .

On considère un modèle de régression linéaire de  $\mathbb{R}^p$  dans  $\mathbb{R}$ .

Soit un n-échantillon  $(x_1, y_1), \dots, (x_n, y_n) \in \mathbb{R}^p \times \mathbb{R}$  indépendant et identiquement distribué que l'on suppose généré suivant le modèle  $Y = \beta^T X + \varepsilon$  où  $\varepsilon \sim N(0, \sigma^2)$ .

Vérifiez que, conditionnellement à  $X$ ,  $Y$  suit une loi Gaussienne de moyenne  $\mu(X)$ , que l'on précisera, et de variance  $\sigma^2$ .

**Question 3.b (1 point)**

Quels sont les estimateurs du maximum de vraisemblance de la variance  $\sigma^2$  et de  $\beta$  dans ce cas ?

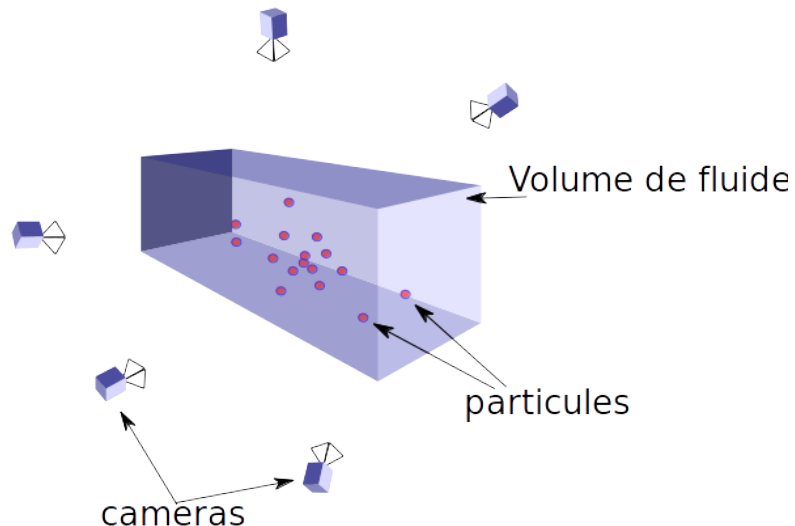
**Exercice 4 (programmation – 4 points)**

Une équipe Inria travaille sur l'analyse d'images pour extraire des paramètres d'écoulement fluides.

Certaines techniques utilisées nécessitent des traitements informatiques lourds. L'équipe a identifié une brique fondamentale de ces traitements qui pourrait être améliorée par l'utilisation des technologies parallèles et fait appel à un ingénieur du SED pour étudier les pistes d'accélération de ces traitements et éventuellement en définissant une architecture matérielle cible adaptée.

### Description du problème :

Une des techniques utilisée pour l'analyse d'écoulement est la capture de mouvement des petites particules, boules d'air ou de savon qui sont portées par l'écoulement. Pour faire cela, une série de caméras enregistre depuis des angles différents les particules se déplaçant dans un volume du fluide.



L'équipement dispose de 5 caméras d'une résolution de 1280 x 800.

Le nombre de particules peut varier énormément d'une expérimentation à une autre. On prend comme référence le nombre de 20000 particules.

Pour définir la position de chaque particule dans le volume, le volume de fluide observé est discrétisé en une grille de 1600 x 900 x 160 soit 230,4 millions de cases. Si la case contient 0, alors la probabilité de présence de la particule est nulle. Plus la valeur dans la case est grande, plus y a de chances que la particule s'y trouve.

La méthode utilisée par l'équipe est de partir d'un ensemble de particules « virtuelles » ou « synthétiques » qu'on projette sur l'espace caméra, le but étant ensuite d'optimiser la position et le nombre des particules pour coller à la vérité terrain.

### Algorithme de projection des particules sur l'espace des images caméra :

On simule le résultat d'une prise de vue par les 5 caméras d'un ensemble  $L$  de particules. Le résultat est une accumulation des poids de chaque particule sur l'espace de pixels de l'ensemble de 5 caméras.

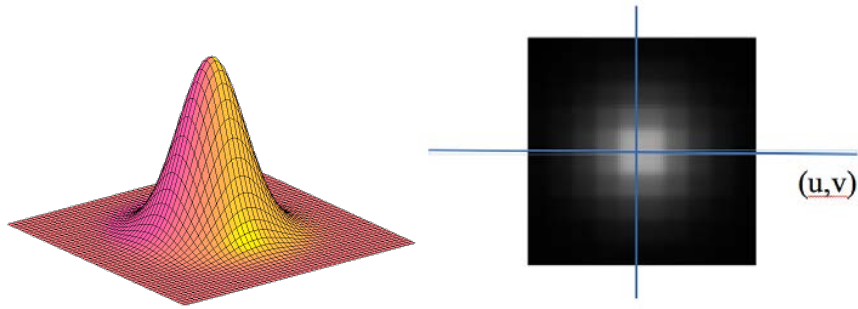
#### Pour une particule donnée :

On dispose d'une fonction projection qui retourne les coordonnées  $(u,v)$  dans le plan de la caméra  $i$  du centre de la projection d'une particule à partir de ses coordonnées  $(x,y,z)$  dans le volume de fluide.

$$(u, v) = projection((x, y, z), i)$$

La complexité de la fonction projection est de l'ordre de 20 opérations arithmétiques (+, \*, /) sur des nombres flottants.

On applique à partir du point  $(u,v)$  de la projection une fonction gaussienne centrée en  $(u,v)$  sur environ 10 x10 pixels de l'image. Voir la représentation ci-dessous de la fonction gaussienne en 3 dimensions et celle de l'image (en pixels) d'une caméra :



La fonction gaussienne est définie par la formule :

$$gaussian(u_p, v_p, u, v) = Ae^{-\frac{(u_p-u)^2+(v_p-v)^2}{2\sigma^2}}$$

Où A et  $\sigma$  sont des constantes et  $(u_p, v_p)$  sont les coordonnées d'un pixel dans image d'une caméra. On l'applique sur un voisinage de 10 x 10 pixels autour de la position  $(u, v)$ .

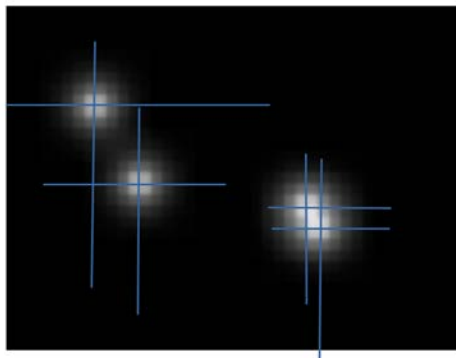
Voici le pseudo-code du calcul de la projection d'une particule sur le plan Image d'une caméra i :

```
(u,v) <- projection(x,y,z,i)
pour (u_p,v_p) dans voisinage((u,v), 10, 10) faire :
    Image( i, u+u_p, v+v_p) <- Image( i, u+u_p, v+v_p) + gaussian( u_p, v_p, u, v )
```

On obtient alors une nouvelle image synthétique ressemblant à l'image capturée par la caméra i pour une particule se trouvant dans la position  $(x, y, z)$ .

Pour toutes les particules :

Maintenant il faut accumuler l'ensemble des contributions de toutes les particules sur les 5 plans image. Le résultat sera 5 images synthétiques simulant les prises de vue des 5 caméras. Pour réaliser ces images, on additionne les différentes gaussiennes produites par chaque particule sur chaque image caméra. Le résultat pour une image ressemble à ceci (avec les coordonnées de 4 particules) :



**Question 1 (1 point)**

Écrivez en pseudo-code l'algorithme qui prend en entrée une liste de particules L et rend en sortie les 5 images synthétiques des 5 caméras.

### Question 2 (1/2 point)

Calculez la taille en mémoire des données nécessaires pour représenter :

- l'ensemble des positions des particules (codés en float 32),
- l'ensemble de cases du volume du fluide (codés en float 32),
- les images résultantes des 5 caméras (codés en float 32).

### Question 3 (1 point)

Que préconisez-vous pour paralléliser le traitement de cette brique ? Donnez un pseudo-code modifié en fonction de vos préconisations.

### Question 4 (1 point)

Compte tenu de vos précédentes réponses, quel type d'architecture serait le meilleur pour réaliser ce calcul en parallèle ? Pouvez-vous justifier votre choix ?

### Question 5 (1/2 point)

Les chercheurs ont besoin de visualiser leurs résultats. Que proposez-vous comme bibliothèques d'affichage ?

## Exercice 5 (apprentissage – 4 points)

Une équipe de recherche a développé une application mobile de mesure de la pollution sonore. Des relevés du bruit ambiant sont pris à intervalles réguliers par l'application (niveau dB(A) calculé sur le téléphone à partir d'extraits sonores de 5 secondes, complété par des informations de différents capteurs du téléphone) et envoyés à un serveur.

Les micros des téléphones mobiles sont cependant limités, et surtout ils ne sont pas calibrés d'origine. Le niveau de dB(A) calculé ne correspond donc pas au niveau de dB(A) réel. Un biais doit être appliqué aux mesures prises pour avoir le niveau réel de bruit ambiant.

Pour une mesure  $i$ , on note  $X_i$  le niveau sonore réel. Si celui-ci ne se situe pas dans la zone de saturation du téléphone (bruit trop faible ou trop élevé), on modélise le niveau sonore  $Y_i^t$  mesuré par le téléphone  $t$  par l'équation

$$Y_i^t = X_i + \text{biais}^t + \varepsilon_i^t$$

où

- $\varepsilon_i^t$  est un bruit indépendant et identiquement distribué de moyenne 0
- $\text{biais}^t$  est le biais pour le téléphone  $t$

Des sessions de calibration (étalonnage des téléphones avec un sonomètre professionnel par comparaison avec un bruit rose à différents niveaux sonores) ont permis d'estimer le biais pour une centaine de modèles, dont les principaux modèles vendus. Suite à ces expérimentations (Figure 1.a, 1.b, 1.c), il est conclu que :

- les téléphones mobiles commencent à saturer dans un environnement entre 75 et 80 dB(A) (niveau de bruit réel).

- le micro des téléphones mobiles est peu ou pas activé en dessous de 40 dB(A) (niveau de bruit réel).
- le biais à appliquer est généralement constant sur l'intervalle 40 dB(A) - 75 dB(A).
- le biais calculé sur un téléphone peut généralement être utilisé pour les téléphones du même modèle.

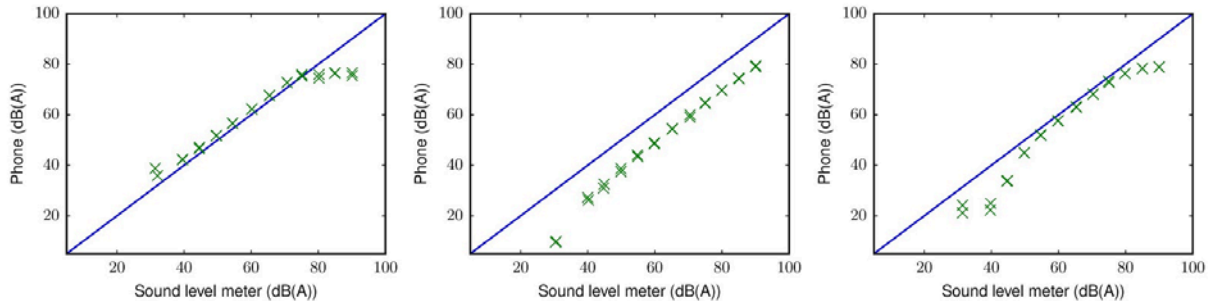


Figure 1: (a, b, c): rapport sonomètre/téléphone du niveau sonore détecté pour 3 téléphones.

Ces sessions de calibration ne permettent cependant pas de calibrer beaucoup de téléphones (plusieurs centaines de nouveaux modèles sont sortis chaque année par les fabricants). Des solutions alternatives d'auto-calibration sont donc envisagées. L'équipe souhaite entre autres explorer l'utilisation du Machine Learning pour cela.

Téléphone	Biais estimé (dB(A))
LGE Nexus 5 / 4	-6.12 / -0.58
Samsung GT I9505 / I9300 / I9305 / I9250 / I9195	-10.97 / -7.16 / -3.32 / -2.51 / -6.02
Samsung SM C115 / A500H / A500FU	-18.97 / -6.8 / -7.7
OnePlus A0001	-14.75
LGE LG-D802 / LG-D855 / LG-H815	0.0 / -6.52 / -24.53
HUAWEI U8860 / P6-U06 / Y300-0100	-17.65 / -12.6 / -14.97
Sony C5303 / D5803 / D6503 / D6603 / LT22i	-20.73 / -10.37 / -20.9 / -7.89 / -18.37
HTC HTC One mini 2 / HTC HTC One X	-11.8 / -19.73

Il a été remarqué que :

- A partir d'un certain nombre de mesures, la distribution des mesures (en %) se stabilise. Pour cet exercice, on considérera que la distribution est stable à partir de 10000 mesures.
- La distribution des mesures de niveau sonore est généralement identique pour différents utilisateurs ayant le même modèle de téléphone (une fois stabilisée) (cf Figures 2 et 3)
- La distribution est différente entre différents modèles/constructeurs, même si la forme générale (pic initial, bosse) reste la même (cf. Figure 4)

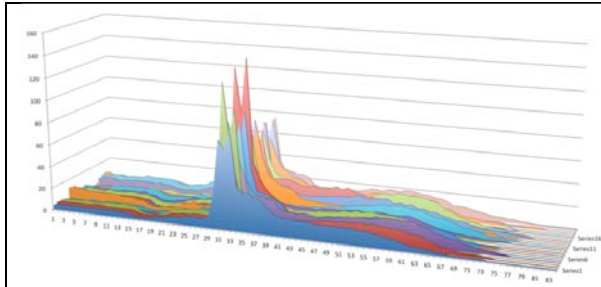


Figure 2: Distribution (%) du niveau sonore brut (raw SPL) en dB(A) pour 20 téléphones Samsung SM-G901F

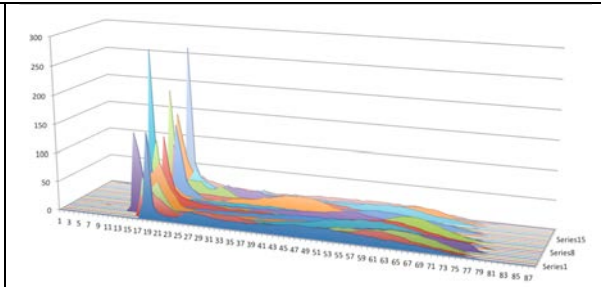


Figure 3: Distribution (%) du niveau sonore brut (raw SPL) en dB(A) pour 20 téléphones Samsung GT-i9305

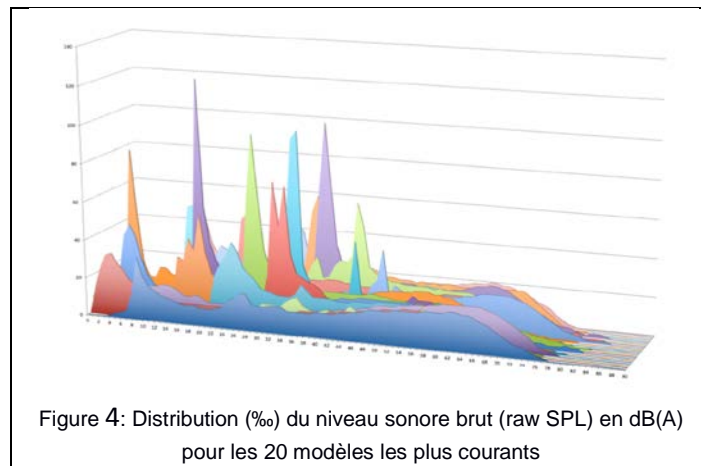


Figure 4: Distribution (%) du niveau sonore brut (raw SPL) en dB(A) pour les 20 modèles les plus courants

### Données d'entrée

Vous disposez du tableau des biais pour la centaine de téléphones calibrés (identificateur unique de modèle, valeur du biais).

Vous disposez également d'une matrice  $A$  de taille  $m \times n$  où

- $m$  est le nombre total de téléphones ayant envoyé des mesures (environ 10000)
- en colonne 0 est donné un identificateur unique de téléphone
- pour les colonnes 1 à 90, le nombre de mesures prises par le téléphone avec le niveau de dB(A) équivalent (de 1 à 90)
- en colonne 91 un identifiant unique de modèle de téléphone.

### **Question 1 (1 point)**

Quel(s) algorithme(s) proposez-vous d'utiliser pour mettre au point une fonctionnalité d'auto-calibration.

### **Question 2 (1/2 point)**

Au vu des données d'entrée, l'(les) algorithme(s) envisagé(s) est(sont)-il(s) stable(s) ?

### **Question 3 (1/2 point)**

Suivant l'algorithme envisagé, proposez une méthode estimant sa précision.



**Question 4 (1/2 point)**

Des transformations des données d'entrée vous semblent-elles potentiellement utiles ou nécessaires à faire dans le cadre de certains algorithmes ? Si oui, lesquelles, pourquoi et pour quel(s) algorithme(s).

**Question 5 (1/2 point)**

Combien de temps vous semble-t-il nécessaire pour réaliser une première étude/évaluation de l'utilité du Machine Learning pour cette problématique ? Détaillez un plan de travail général.

**Question 6 (1 point)**

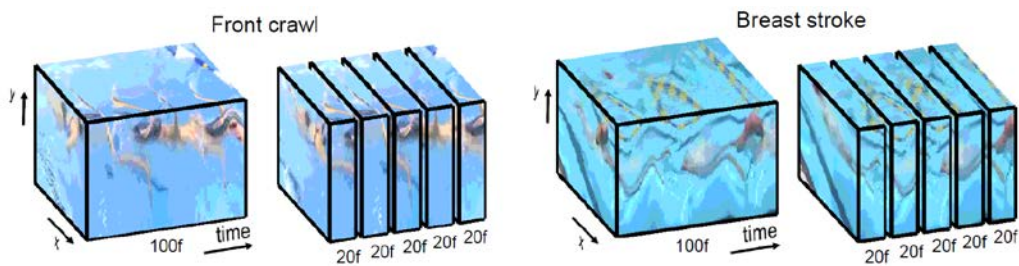
Proposez l'implémentation d'un des algorithmes en pseudo code, et indiquez les bibliothèques et fonctions que vous utilisez dans le langage de votre choix.

## Learning from Synthetic Humans



Estimating human pose, shape, and motion from images and video are fundamental challenges with many applications. Recent advances in 2D human pose estimation use large amounts of manually-labeled training data for learning convolutional neural networks (CNNs). Such data is time consuming to acquire and difficult to extend. Moreover, manual labeling of 3D pose, depth and motion is impractical. In this work we present SURREAL: a new large-scale dataset with synthetically-generated but realistic images of people rendered from 3D sequences of human motion capture data. We generate more than 6 million frames together with ground truth pose, depth maps, and segmentation masks. We show that CNNs trained on our synthetic dataset allow for accurate human depth estimation and human part segmentation in real RGB images. Our results and the new dataset open up new possibilities for advancing person analysis using cheap and large-scale synthetic data.

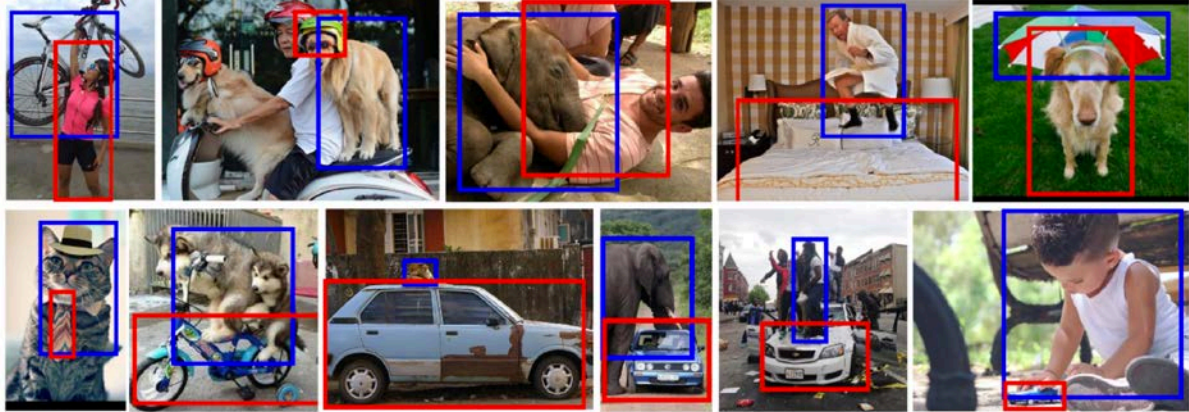
## Long-term Temporal Convolutions for Action Recognition



Typical human actions last several seconds and exhibit characteristic spatio-temporal structure. Recent methods attempt to capture this structure and learn action representations with convolutional neural networks. Such representations, however, are typically learned at the level of a few video frames failing to model actions at their full temporal extent. In this work we learn video representations using neural networks with long-term temporal convolutions (LTC). We demonstrate that LTC-CNN models with increased temporal extents improve the accuracy of action recognition. We also study the impact of different low-level representations, such as raw values of video pixels and optical flow vector fields and demonstrate the importance of high-quality optical flow estimation for learning accurate

action models. We report state-of-the-art results on two challenging benchmarks for human action recognition UCF101 (92.7%) and HMDB51 (67.2%).

## Weakly-supervised learning of visual relations



This paper introduces a novel approach for modeling visual relations between pairs of objects. We call relation a triplet of the form (subject, predicate, object) where the predicate is typically a preposition (eg. 'under', 'in front of') or a verb ('hold', 'ride') that links a pair of objects (subject, object). Learning such relations is challenging as the objects have different spatial configurations and appearances depending on the relation in which they occur. Another major challenge comes from the difficulty to get annotations, especially at box-level, for all possible triplets, which makes both learning and evaluation difficult. The contributions of this paper are threefold. First, we design strong yet flexible visual features that encode the appearance and spatial configuration for pairs of objects. Second, we propose a weakly-supervised discriminative clustering model to learn relations from image-level labels only. Third we introduce a new challenging dataset of unusual relations (UnRel) together with an exhaustive annotation, that enables accurate evaluation of visual relation retrieval. We show experimentally that our model results in state-of-the-art results on the visual relationship dataset significantly improving performance on previously unseen relations (zero-shot learning), and confirm this observation on our newly introduced UnRel dataset.

## Annexe B: Proposition de Projet MDLearn NG

### Context

Although tools for data analysis have improved, data scientists still spend large amount of time to manipulate data, and evaluate data quality issues. Solving quality issues often involves regular reformatting of data values and layout, correction of erroneous values and missing values, and integration of multiple data sources. These transformations are often difficult to identify using analytical tasks.

To address the exact same issues he faced during his PhD work, Anthony developed a Web platform, MDLearn, to visualize, compare, transform Machine Learning (ML) datasets he received from various partners in a collaborative research project in a graphical, interactive way. ML datasets visualization and manipulation enables to some extent the early detection of problems such as classifier failure, outliers, or issues with data distribution. It also enables the detection of potential new signals for ranking.

For data cleaning, the main open source alternative to MDLearn is OpenRefine (formerly Google Refine), an open source project for cleaning datasets, transforming them from one format into another; and extending them with web services and external data. This project has however seen reduced activity, and is mostly focused on text-oriented datasets. It also had performance issues for memory intensive datasets.

Data Wrangler was another Open Source alternative from Stanford Visualization Group but the software is no longer actively supported, and instead a commercial solution (Trifacta) has been developed from this project.

For interactive data visualization, many libraries (e.g., D3js, leaflet and many more) using JavaScript and HTML5 have been developed to render charts, networks, or hierarchical representations, and considerable progress has been made in integrating and displaying highly complex data. These library however focus on the visualization of data already processed and cleaned up, They are not dealing with data quality issues. Furthermore, these libraries lack the semantic of ML oriented datasets (e.g., comparing training vs sample data sets).

Starting from scratch, MDLearn now represents about 4000 lines of Python code for the backend, and 4500 lines of JavaScript code, 1500 lines of CSS3 and 1250 lines of HTML for the front-end, developed following an agile approach. Multiple open source libraries have been used to parse various input data streams and files, manipulate complex hierarchical data models, and display customized UI components.

As the number of input and data format increases, along with the size of the data sets, the current implementation of MDLearn do exhibits performance issues. A new flexible architecture supporting dynamic plugins is required, as well as performance optimizations.

### Objectives

- Refactor MDLearn with a flexible plugin architecture for data input and output format, transformers, and visual interactions.
- Port all supported formats and transformers as plugins in the new architecture.
- Release as Open Source software, Initiate a developer community around the platform to sustain its maintenance and evolution.

### Workplan

#### TASK 0: Project Management

Duration	18 months
----------	-----------

Delivery	T0+18
Resources	<ul style="list-style-type: none"> <li>- Marc (0.5 month)</li> <li>- Anthony (0.5 month)</li> <li>- Engineer</li> </ul>
Objectives	<ul style="list-style-type: none"> <li>- Project planning and resource management</li> </ul>
Actions	<ul style="list-style-type: none"> <li>- Bi-monthly meeting for planning and monitoring of development tasks.</li> </ul>

### TASK 1: MDLearn NG platform architecture

Duration	2 months
Delivery	T0+2
Resources	<ul style="list-style-type: none"> <li>- 1 engineer (2 months)</li> <li>- Anthony (1 month)</li> <li>- Marc (0.5 month)</li> </ul>
Objectives	<ul style="list-style-type: none"> <li>- Getting to know the existing platform</li> <li>- Define the new core and plugin architecture</li> </ul>
Actions	<ul style="list-style-type: none"> <li>- Learn about current architecture and code</li> <li>- Assess the validity of the current choices (Python, Angular2, and UI component javascript libraries)</li> <li>- Migrate current project under Inria's Gitlab and put project under Inria's continuous integration platform.</li> <li>- Define plugin architecture for input sources and data visualisation components</li> </ul>

### TASK 2 - Core refactoring and plugin subsystem development

Duration	2 months
Delivery	T0+4
Resources	<ul style="list-style-type: none"> <li>- 1 engineer (2 months)</li> <li>- Anthony (1 month)</li> </ul>
Objectives	
Actions	

### TASK 3 - Plugins development

Duration	6 months
Delivery	T0+10
Resources	<ul style="list-style-type: none"> <li>- 1 engineer (6 months)</li> <li>- Anthony (0.5 month)</li> </ul>
Objectives	<ul style="list-style-type: none"> <li>- Port existing input/output data processing units and data visualization components as plugins in the new architecture</li> </ul>
Actions	<ul style="list-style-type: none"> <li>- Port existing input/output data processing units and data visualization components (no support for interactive transformation at this stage)</li> <li>- Develop unit tests as well as CI jobs for testing on various Web browser configurations</li> </ul>

### TASK 4 - MDLearn NG Internal Release

Duration	2 months
Delivery	T0+12
Resources	<ul style="list-style-type: none"> <li>- 1 engineer (2 months)</li> </ul>

Objectives	<ul style="list-style-type: none"> <li>- finalize the integration of the plugins</li> <li>- build and deploy for internal use</li> </ul>
Actions	<ul style="list-style-type: none"> <li>- finalize the integration of the plugins</li> <li>- packaging and release process</li> <li>- fixbugs</li> <li>- initial developer and user documentation</li> <li>- deploy on test platform</li> </ul>

### TASK 5 - Interactive transformation

Duration	4 months
Delivery	T0+16
Resources	<ul style="list-style-type: none"> <li>- 1 engineer (4 months)</li> </ul>
Objectives	<ul style="list-style-type: none"> <li>- Support interactive transformation of ML Datasets</li> </ul>
Actions	<ul style="list-style-type: none"> <li>- Add UI components and management for workflow for data transformation</li> <li>- unit testing</li> <li>- deploy on test platform</li> </ul>

### Task 6: Open Source Release

Duration	2 months
Delivery	T0+18
Resources	<ul style="list-style-type: none"> <li>- 1 engineer (2 months)</li> <li>- Anthony (0.5 month)</li> <li>- Marc (0.5 month)</li> </ul>
Objectives	<ul style="list-style-type: none"> <li>- Public open source release and user support</li> </ul>
Actions	<ul style="list-style-type: none"> <li>- Finalize user documentation and demo application</li> <li>- Write a set of how-to tutorials</li> <li>- Define process for interacting with community/external contributors</li> <li>- Publish as open source project on relevant platform (GitHub,...) to define with the Transfer and Innovation service (also decide OSS license being used).</li> <li>- Provide assistance for the integration of new plugins</li> <li>- Fixbugs</li> </ul>

## Evaluation

#### Platform/code indicators

- Public release of the MDLearn NG
- Code coverage and unit tests
- Performance tests for large data sets

#### Diffusion/acceptance indicators

- Activity on the project's public repository (issues from external people, external contributions, contact request)
- Use of MDLearn NG by external projects