



# **Syllabus S8**

## **Majeure**

## **Image**

**Promo 2026**

## PYBD - Python Big Data

- Niveau : M1
- Semestre : SI8
- Coefficient UE : ?
- Volume horaire total : 12
- Volume face-à-face (TOTAL / EQ-CM / RMD / TDP) : 12 / 0 / 0 / 12
- Langue : Fr
- Référent : Olivier Ricou

## Pré-requis

Connaitre Python et Numpy.

## Résumé et contexte

Apprendre à traiter les données massives avec Python + Pandas, puis à les visualiser avec différentes bibliothèques de Python (Matplotlib, Seaborn, Plotly,...).

## Acquis d'apprentissages visés (AAVs)

Il est attendu qu'un élève sache après ce cours

- nettoyer un jeu de données massives
- réordonner les données
- analyser les données
- faire des graphiques de tout type, 2D, 3D, statique ou dynamique

## Plan cours

Introduction sur les enjeux des données massives.

### Pandas

- Pandas indexing
- View vs copy
- Cleaning a dataframe
- N-dimensions dataframe or multi-index
- Group data
- Merging 2 dataframes
- Time dataframe
- Tools
- Getting data

## Graphics

- Static graphics with Matplotlib
- Static graphics with Seaborn
- Dynamic graphics with Plotly – Basics
- Dynamic graphics with Plotly – Scatter and bubbles
- Dynamic graphics with Plotly – More chart types
- Dynamic graphics with Plotly – Subplots
- Dynamic graphics with Plotly – 3D
- Maps
- A dashboard with Dash – Layout
- A dashboard with Dash – Events

## Format des activités

- TP
- Projet

## Évaluation des AAVs

Evaluations par un projet.

## Calcul de note finale

Note du projet.

## Références et bibliographie

- <https://python3.mooc.lrde.epita.fr/tree/> Leçons 6 et 7
- <https://delta.lrde.epita.fr/> comme exemple de ce qui est attendu pour le projet

## **IREN - Introduction aux réseaux neuronaux**

- Niveau : M1
- Semestre : SI8
- Coefficient UE : ?
- Volume horaire total : 12
- Volume face-à-face (TOTAL / EQ-CM / RMD / TDP) : 12 / 6 / 0 / 6
- Langue : Fr
- Référent : Olivier Ricou

### **Pré-requis**

Connaitre Python, Numpy et Pandas.

### **Résumé et contexte**

Introduction aux réseaux neuroneaux avec un focus sur les réseaux à convolution (CNN) pour classer des images.

### **Acquis d'apprentissages visés (AAVs)**

Il est attendu qu'un élève sache après ce cours

- préparer les données nécessaire à un CNN
- construire un réseau neuronal dense et convolutif
- analyser ses résultats
- utiliser l'environnement Keras et Kaggle

### **Plan cours**

- Tour d'horizon
- Présentation de la théorie des réseaux neuronaux
- Expérimentation sur un séparateur
- Études de CNN
- Présentation du projet sous forme de concours Kaggle

### **Format des activités**

- Cours magistral
- TP
- Projet

### **Évaluation des AAVs**

Evaluations par un projet.

## **Calcul de note finale**

Note du projet.

## **Références et bibliographie**

[http://www.ricou.eu.org/iren/notes\\_rn.html](http://www.ricou.eu.org/iren/notes_rn.html)

## VITK – Introduction à ITK et VTK

Niveau : M1	Semestre : S8	Coefficient : 1
Volume horaire total : 26	Volume horaire face-à-face : 14	8 / 0 / 6 / 0 (CM/TD/TP/Examen)
Langues : Français	Enseignant référent : Roman Fenioux (Kitware)	

### Prérequis

- Aucun

### Résumé du cours

Dans le domaine du traitement d'images médicales, la manipulation et la visualisation d'images en 3D sont une problématique complexe. Les bibliothèques ITK et VTK sont particulièrement adaptées, mais leur utilisation peut être compliquée pour des non-initiés. Kitware, développeur de ces bibliothèques, est le mieux placé pour former les étudiants à ces bibliothèques particulières.

### Contenu et plan du cours

- Introduction à VTK
- Introduction à ITK
- Application au recalage d'images médicales

### Format des activités

Formats des activités pédagogiques :

- Cours magistraux
- Travaux pratiques
- Projet

### Acquis d'apprentissage attendus

A l'issue de ce cours, les étudiants sont capables :

- De manipuler efficacement les bibliothèques VTK et ITK
- De faire des visualisations et du recalage d'images médicales en 3 dimensions

### Évaluation des acquis d'apprentissage attendus

- Un projet mettant en pratique les notions vues en cours et en TP

### Formule de calcul de la note finale

100% projet

### Références et bibliographie

- *Documentation des bibliothèques ITK et VTK*

## TIFO 1 – Traitement d'Images Fondamental

Niveau : M1	Semestre : S8	Coefficient : 1,5
Volume horaire total : 45	Volume horaire face-à-face : 30	15/ 0 / 15 / 0 (CM/TD/TP/Examen)
Langues : Français	Enseignant référent : Jonathan Fabrizio	

### Prérequis

- Bases de C++

### Résumé du cours

Avant de pouvoir réaliser des algorithmes de reconnaissance des formes sur des images ou mettre en œuvre de l'IA sur des photos, il est nécessaire de connaître et maîtriser les bases de la formation de l'image et du traitement d'images. L'objectif du cours est de présenter aux étudiants les briques élémentaires incontournables de traitement d'images, permettant l'élaboration d'algorithmes plus complexes.

Ce cours d'introduction au traitement d'images vise à familiariser les étudiants avec la terminologie et les traitements élémentaires qui structurent les algorithmes avancés de restauration, amélioration, filtrage et extraction automatique d'information des images. Les notions abordées balayent la formation des images (perception et capteurs), le codage (niveaux de gris, espaces colorimétriques), la représentation des images (histogrammes et leur manipulation, représentations spatiales et fréquentielles - TF2D), le filtrage. Une ouverture sur les implémentations rapides (instructions SIMD) sera effectuée et servira d'introduction au cours de programmation GPGPU.

### Contenu et plan du cours

- Introduction et histoire du traitement d'images
- Formation de l'image, perception, capteurs,
- Codage et Représentations
- Espaces colorimétriques
- Histogrammes
- Domaines spatial et fréquentiel/TF2D, filtrage, débruitage.

### Format des activités



Formats des activités pédagogiques :

- Cours magistraux
- Travaux pratiques
- Projets

### Acquis d'apprentissage attendus

A l'issue de ce cours, les étudiants sont capables :

- d'expliquer le codage des images.
- de choisir le bon espace de couleur en fonction de la problématique posée
- de développer des filtres pour une chaîne de traitement d'images
- de choisir une stratégie pour réduire le bruit dans une image.

### Évaluation des acquis d'apprentissage attendus

- Durant certaines séances, un QCM sans document et à points négatifs d'environ 10 minutes (vérification directe de la connaissance du cours)
- Un projet commun avec TIFO 2 où l'on développe un programme manipulant les images touchant à un ou plusieurs thèmes :
  - a. du filtrage (morphologie mathématique, dans le domaine spatial ou dans le domaine fréquentiel)
  - b. problématique liée à la couleur
  - c. suivi d'objets
- Les TP ne sont pas notés mais peuvent entraîner des pénalités s'ils sont négligés.

### Formule de calcul de la note finale

$(3 \times \text{Note projet} + \text{Note QCM}) / 4 - \text{pénalités}$

Pénalités : Le fait de bâcler un TP, d'avoir des absences injustifiées ou tout autre manquement aux consignes données tout au long du cours peuvent entraîner des pénalités.

### Références et bibliographie

- « *Digital Image Processing* », Gonzalez et Woods

- « *Digital Image Processing* », K. R. Castleman
- « *The Image Processing Handbook* », J. C. Russ

## TIFO 2 – Traitement d’Images Fondamental 2

Niveau : M1	Semestre : S8	Coefficient : 1
Volume horaire total : 30	Volume horaire face-à-face : 18	9 / 0 / 9 / 0 (CM/TD/TP/Examen)
Langues : Français	Enseignant référent : Elodie Puybareau	

### Prérequis

- TIFO 1

### Résumé du cours

Dans la continuité de TIFO 1, l’objectif du cours est de présenter aux étudiants les briques élémentaires incontournable de traitement d’images, avec une emphase sur la manipulation des différents outils en python.

Ce cours complète les bases de TIFO 1 avec d’autres techniques et notions indispensables dans le domaine du traitement d’images. Cette partie s’intéresse à la reconnaissance de formes simples (droites, cercles), la morphologie mathématique, les méthodes de traitement de séquences d’images, information temporelle, la segmentation et l’évaluation des résultats.

### Contenu et plan du cours

- Introduction à la morphologie mathématique
- Segmentation, débruitage et évaluation
- Approches 3D, 2D+t et 3D+t
- Recalage, tracking et descripteurs

### Format des activités

Formats des activités pédagogiques :

- Cours magistraux
- Travaux pratiques
- Projet commun avec TIFO 1

### Acquis d’apprentissage attendus

A l'issue de ce cours, les étudiants sont capables :

- De mettre en place une pipeline de traitement d'image répondant à une problématique donnée
- de mettre en œuvre des techniques de morphologie mathématique pour résoudre un problème
- d'utiliser les méthodes de segmentations adéquates en fonction d'une problématique définie
- de recalibrer des images 2D, 3D et d'étendre au 2D+t
- d'exploiter les informations temporelles d'une séquence d'images
- Valider les résultats d'une chaîne de traitement d'images en utilisant les métriques appropriées

### Évaluation des acquis d'apprentissage attendus

- Un projet commun avec TIFO 1 dont le but est de développer une chaîne de traitement d'image répondant à une problématique, qui peut toucher un ou plusieurs thèmes parmi :
  - a. du filtrage (morphologie mathématique, dans le domaine spatial ou dans le domaine fréquentiel)
  - b. problématique liée à la couleur
  - c. suivi temporel
- Les TP ne sont pas notés mais peuvent entraîner des pénalités s'ils sont négligés.

### Formule de calcul de la note finale

100% projet – pénalités le cas échéant

### Références et bibliographie

- « *Digital Image Processing* », Gonzalez et Woods
- « *Digital Image Processing* », K. R. Castleman
- « *The Image Processing Handbook* », J. C. Russ
- « *Morphologie mathématique* » tome 1 et 2, Talbot et Najman

## POGL – Programmation OpenGL

Niveau : M1	Semestre : Printemps	Coefficient : X
Volume horaire total : 40	Volume horaire face-à-face : 21	6 / 0 / 15 / ?? (CM/TD/TP/Examen)
Langues : Français	Enseignant référent : Jonathan Fabrizio	

### Prérequis

- Avoir suivi le cours « Introduction à la synthèse d'images (ISIM) en ING2 », et avoir acquis les concepts de base de la synthèse temps réel (z-buffer, backface culling...),
- Être capable de programmer en C++,
- Maîtriser les notions de bases de géométrie Euclidienne et la représentation des transformations usuelles (translation, rotation...) par des matrices homogènes comme cela est rappelé en cours d'ISIM.

### Résumé du cours

Fournir les bases pour utiliser OpenGL 4 ;

Le premier objectif est de maîtriser les rudiments programmation classique d'OpenGL et la programmation des shaders en utilisant GLSL. On prendra bien soin de viser les versions actuelles d'OpenGL et de jamais utiliser des éléments des versions antérieurs.

Une fois ce premier objectif atteint, le second objectif sera d'apprendre à réaliser des rendus plus difficiles en temps réel tel que le rendu d'eau, de feu, d'éléments réfléchissants ou la gestion des ombres en temps réel. Cela sera l'occasion d'aborder des techniques comme le render to texture, le multiple render targets...

Des méthodes classiques pour accélérer le rendu seront abordés comme le deferred rendering. Tout au long du cours nous nous appuierons sur des algorithmes de rendu temps réel présentés dans le cours ISIM.

### Contenu et plan du cours

- Introduction à OpenGL
- Comment construire un programme en OpenGL
- Pipeline graphique et programmation de *shaders* avec GLSL

- Les différents repères dans la formation d'une image avec OpenGL.
- Effets temps reels et méthodes de rendu avancées :  
FBO/*Render to texture* (rendu d'ombre, rendu d'eau)

Feu/particules

*Environnement map*

...

- Accélération de rendu temps réel (optimisation : *deferred rendering*)

## Format des activités

Formats des activités pédagogiques :

- Cours magistraux
- Travaux pratiques
- Projets

## Acquis d'apprentissage attendus

A l'issue de ce cours, les étudiants sont capables :

- D'utiliser l'API d'OpenGL ce qui comprends :
  - a. paramétrer le contexte d'OpenGL
  - b. compiler et lier un programme écrit en GLSL
  - c. initialiser des VBO et VAO
  - d. lancer l'envoi des primitives graphiques
  - e. éventuellement utiliser des textures.
- Programmer en GLSL un vertex shader et un fragment shader
- Utiliser des bibliothèques utiles pour interagir avec l'utilisateur et le contexte OpenGL (comme glut, glew).

## Évaluation des acquis d'apprentissage attendus

- A chaque séance de TP ou presque, un QCM sans document et à points négatifs d'environ 10 minutes (vérification directe de la connaissance du cours)
- Un projets ou l'on développe un programme reposant sur OpenGL faisant intervenir des shaders GLSL.
- Les Tps ne sont pas noté mais peuvent entraîner des pénalités s'ils sont négligés.

## Formule de calcul de la note finale

(3\*Note projet+Note QCM) / 4 – pénalités

Pénalités : Le fait de bâcler un TP, d'avoir des absences injustifiées ou tout autre manquement aux consignes données tout au long du cours peuvent entraîner des pénalités.

### Références et bibliographie

- [www.opengl.org](http://www.opengl.org)
- « OpenGL Programming Guide », Dave Shreiner, Graham Sellers, John Kessenich, Bill Licea-Kane

## OCVX1 – Optimisation Convexe

Niveau : M1	Semestre : S8	Coefficient : <b>FIXME</b>
Volume horaire total : 64	Volume horaire face-à-face : 32	18/8/6/2 (CM/TD/TP/Examen)
Langues : Français	Enseignant référent : TOCHON Guillaume	

### Prérequis

- Les bases d’algèbre linéaire et de calcul matriciel : valeurs/vecteurs propres d’une matrice, définition d’une matrice définie positive, etc
- Les bases de géométrie euclidienne : vecteurs et espaces vectoriels, produit scalaire, norme, écriture paramétrique/implicite d’une droite, d’un (hyper)plan, etc
- Les bases d’analyse fonctionnelle : continuité, dérivabilité d’une fonction univariée, gradient d’une fonction multivariée.

### Résumé du cours

L'optimisation est une activité de l'ingénieur à part entière ; on cherche à optimiser le rendement d'un processus industriel, à minimiser un coût ou à réduire un risque. Plus récemment les démarches d'optimisation sont apparues au cœur des questions d'apprentissage automatique. En particulier, la problématique d'*apprentissage supervisé* se formule comme un problème d'optimisation : on cherche les meilleurs paramètres d'un modèle donné pour que ce dernier commette le moins d'erreurs possible. De nombreuses tâches en traitement d'image se formulent également comme la solution d'un problème d'optimisation (par exemple, débruiter une image équivaut à supprimer un *maximum* de bruit en dégradant au *minimum* le contenu de l'image).

Le cours d'OCVX est une **initiation aux techniques mathématiques** nécessaires à la compréhension des différents **algorithmes de résolution de problèmes d'optimisation**.

### Contenu et plan du cours

- Introduction et enjeux de l’optimisation convexe – 2h CM
- Rappels et préliminaires géométriques – 3h CTD (2h CM + 1h TD)
- Convexité à l’ordre 0 – 3h CTD (2h CM + 1h TD)
- Les fondements mathématiques de l’optimisation convexe
  - Un peu de calcul différentiel – 3h CM + 3h TD
  - Convexité à l’ordre 1 et à l’ordre 2 et points critiques – 3h CM + 3h TD
- Les algorithmes de descente – 6h CM + 6h TP
- Les méthodes de régularisation – 4h TP

### Format des activités

Formats des activités pédagogiques :

- Cours magistral



- Travaux dirigés
- Travaux pratiques

### Acquis de formation attendus

À la fin du cours, les étudiants sont capables de :

- Identifier les différents éléments composant un problème d'optimisation et des éléments géométriques et analytiques nécessaires à son **étude qualitative**.
- **Cartographier les algorithmes** à disposition pour résoudre un problème d'optimisation et les hyperparamètres qui déclinent et gouvernent ceux-ci.
- Décrire le **domaine de validité** d'un algorithme.
- **Implémenter des algorithmes** d'optimisation sans contraintes.
- Effectuer des **analyses comparatives** entre des différents algorithmes d'optimisation sans contraintes.

### Évaluation des acquis de formation attendus

Évaluation des acquis des notions théoriques par un devoir sur table (2h)

Évaluation des acquis des notions pratiques par compte rendu de TP

### Formule de calcul de la note finale

50 % partiel + 50 % rendu de TP

### Référence et Bibliographie

[https://web.stanford.edu/~boyd/cvxbook/bv\\_cvxbook.pdf](https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf) Le livre *Convex Optimization* de Boyd & Vandenberg : la bible de l'optimisation convexe

Les ressources sur la page Moodle du cours

## KPRAT – Etude de cas pratique

Niveau : M1	Semestre : S8	Coefficient : <b>1</b>
Volume horaire total : 15	Volume horaire face-à-face : 15	2/12/0/0 (CM/TD/TP/Examen)
Langues : Français	Enseignant référent : GERAUD Thierry	

### Prérequis

- Traitement d'Image Fondamental (TIFO) 1 et 2

### Résumé du cours

La recherche bibliographique et étude de l'état de l'art est une étape primordiale dans la vie d'un ingénieur. Ce cours a pour but d'introduire la recherche bibliographique aux étudiants, en leur présentant les bases d'une recherche bibliographique et en leur fournissant des articles provenant d'actes d'une conférence de référence dans le domaine du traitement d'image qu'ils doivent lire et présenter à toute la classe. Cet exercice leur permet de lire un article scientifique, et d'apprendre à en faire une restitution en utilisant à la fois les données de l'article et les connaissances acquises en cours. Les étudiants sont actifs dans le cours en étant impliqués dans la notation des autres présentations.

### Contenu et plan du cours

- Cours d'introduction à la recherche bibliographique 2h CM
- 6 x 2h de présentations d'articles

### Format des activités

Formats des activités pédagogiques :

- Cours magistral
- Travaux dirigés

### Acquis de formation attendus

À la fin du cours, les étudiants sont capables de :

- Faire une recherche bibliographique
- Lire un article scientifique
- Faire une présentation d'un article scientifique

### Évaluation des acquis de formation attendus

Évaluation des acquis par présentation d'un article scientifique

### Formule de calcul de la note finale

100 % présentation

## Référence et Bibliographie

Supports de cours sur la page Moodle

## ISIM – Introduction à la synthèse d'images

Niveau : M1	Semestre : Printemps	Coefficient : X
Volume horaire total : 60	Volume horaire face-à-face : 30	15 / 0 / 15 / ?? (CM/TD/TP/Examen)
Langues : Français	Enseignant référent : Jonathan Fabrizio	

### Prérequis

- Être capable de programmer en C++,

### Résumé du cours

La synthèse d'images est une technique de l'infographie qui consiste en la création d'images numériques par application d'un modèle physique. Cette discipline trouve de multiples applications pratiques, telles que la réalité virtuelle, la modélisation 3D ou encore les effets spéciaux.

Ce cours a pour but de que les étudiants soient capables de maîtriser les algorithmes fondamentaux pour le rendu photoréaliste et le rendu temps réel.

Ce cours introduit les notions de base de la synthèse d'images.

Après des brefs rappels sur la formation de l'image ainsi la géométrie projective et euclidienne, le cours poursuit deux objectifs.

Le premier objectif est d'apprendre les algorithmes fondamentaux pour le rendu photoréaliste (cinéma, simulation, etc). Les algorithmes classiques seront abordés : raycasting, raytracing, pathtracing, pbrgi... Les problèmes classiques et les limitations de ces algorithmes seront étudiés.

Le second objectif est d'apprendre les fondements du rendu temps réel (jeux vidéos). Les algorithmes classiques du pipeline graphique seront étudiés (fenêtrage, remplissage de polygone...).

Des objectifs transverses seront survolés comme la modélisation des formes ou l'utilisation et la manipulation de textures. Un booster blender est aussi organisé :

pour permettre aux étudiants qui ne connaissent pas blender de prendre en main blender et d'avoir un outil puissant afin de pouvoir modéliser une scene ou faire du rendu

pour approfondir blender (un aspect de blender – décidé individuellement en accord avec l'enseignant) dans le cas ou blender est déjà connu.

## **Contenu et plan du cours**

- Introduction et historique de la synthèse d'images
- Géométrie euclidienne et projective, formation de l'image.
- Rendu photoréaliste (raycasting, raytracing, pathtracing, pbrgi...)
- Rendu temps réel (algorithmes de fenêtrage, remplissage...)
- Modélisation d'objets (maillage et lissage de maillages, CSG, révolution/extrusion, carte de hauteurs, L-system).
- Textures.
- Booster Blender

## **Format des activités**

Formats des activités pédagogiques :

- Cours magistraux
- Travaux pratiques
- Projets

## **Acquis d'apprentissage attendus**

A l'issue de ce cours, les étudiants sont capables :

- d'expliquer les algorithmes de bases pour le rendu temps réel ou photoréaliste
- de programmer un moteur simple (pas forcément efficace) permettant la génération d'image soit par projections (en gérant les problèmes d'occultations, de remplissage), soit en développant un moteur de raytracing ou pathtracing.

## **Évaluation des acquis d'apprentissage attendus**

- A chaque séance de TP ou presque, un QCM sans document et à points négatifs d'environ 10 minutes (vérification directe de la connaissance du cours)
- Un projets ou l'on développe un programme touchant soit à
  - a. un problème de rendu temps réel
  - b. un problème de rendu photoréaliste
  - c. un problème de modélisation
- Les TPs ne sont pas notés mais peuvent entraîner des pénalités s'ils sont négligés.

### Formule de calcul de la note finale

$$(3 * \text{Note projet} + \text{Note QCM}) / 4 - \text{pénalités}$$

Pénalités : Le fait de bâcler un TP, d'avoir des absences injustifiées ou tout autre manquement aux consignes données tout au long du cours peuvent entraîner des pénalités.

### Références et bibliographie

- « *Computer Graphics: principles and practice* », Wesley
- « *Algorithmes pour la synthèse d'images et l'animation 3D* », Malgouyres

## IML – Introduction au Machine Learning

Niveau : M1	Semestre : S8	Coefficient : <b>FIXME</b>
Volume horaire total : 17	Volume horaire face-à-face : 17	5/0/12/0 (CM/TD/TP/Examen)
Langues : Français	Enseignant référent : TOCHON Guillaume	

### Prérequis

- Les bases d'algèbre linéaire et de calcul matriciel
- Le vocabulaire de base de l'optimisation convexe

### Résumé du cours

Le *machine learning* est une discipline absolument essentielle aujourd'hui, de part ses applications dans tous les domaines de l'ingénierie. Le succès d'un algorithme de machine learning pour une tâche précise réside dans l'expertise de l'utilisateur à pré-traiter correctement les données en amont, sélectionner le modèle (ou une famille de modèles) le plus pertinent pour la tâche à réaliser, en régler finement les hyperparamètres de manière à éviter le sur-apprentissage, et évaluer rigoureusement les performances qualitatives et quantitatives. Ce cours présente ces différentes étapes sous un angle principalement pratique.

### Contenu et plan du cours

- Séance 0 : Introduction (1h CM)
- Séance 1 : Réduction de dimension (1h CM + 3h TP)
- Séance 2 : Les techniques de clustering (1h CM + 3h TP)
- Séance 3 : L'apprentissage supervisé (1h CM + 3h TP)
- Séance 4 : Fusion d'information (1h CM) + lancement du projet (3h TP)

### Format des activités

Mis à part la séance d'introduction, le format des autres séances présentent en 1h de CM les notions principales d'une grande thématique de *machine learning*, puis l'approfondissent directement de manière pratique via un TP de 3h.

### Acquis de formation attendus

À la fin du cours, les étudiants sont capables de :

- Identifier les analyses et pré-traitements à appliquer sur un jeu de données cible en amont de l'entraînement d'un modèle.
- Reconnaître une tâche de régression d'une tâche de classification dans le cadre de l'apprentissage supervisé, et choisir le ou les modèles pertinents pour traiter cette tâche.
- Régler les hyperparamètres d'un modèle pour les adapter au mieux à un jeu de données et une tâche précise.
- Reconnaître un scénario de sur-apprentissage, et comment l'éviter en pratique.

- Benchmarker les performances de plusieurs algorithmes équivalents pour une tâche cible, et conduire des analyses qualitatives et quantitatives de résultats obtenus.

### **Évaluation des acquis de formation attendus**

Évaluation des acquis théoriques : un QCM Moodle chaque semaine (pas de créneau imposé dans la semaine, les étudiants sont libres de le faire quand ils le souhaitent), portant sur les notions de la séance précédente.

Évaluation des acquis pratiques : un projet en groupe (2 ou 3 personnes)

### **Formule de calcul de la note finale**

20 % QCM + 80 % note projet

### **Référence et Bibliographie**

Polycopié (externe) d'introduction au machine learning :

[http://cazencott.info/dotclear/public/lectures/IntroML\\_Azencott.pdf](http://cazencott.info/dotclear/public/lectures/IntroML_Azencott.pdf)

Le livre de référence (en anglais) du domaine : <https://hastie.su.domains/Papers/ESLII.pdf>

La documentation de scikit-learn : <https://scikit-learn.org/stable/>

La chaîne YouTube StatQuest :

<https://www.youtube.com/channel/UCtYLUtgS3k1Fg4y5tAhLbw>

Les ressources sur la page Moodle du cours



## IMEDI – Introduction à l'imagerie médicale

Niveau : M1	Semestre : S8	Coefficient : 1
Volume horaire total : 30	Volume horaire face-à-face : 18	9 / 0 / 9 / 0 (CM/TD/TP/Examen)
Langues : Français	Enseignant référent : Elodie Puybareau	

### Prérequis

- TIFO 2

### Résumé du cours

Le but du cours est d'initier les étudiants à l'imagerie médicale, en leur donnant les bases de la physique d'acquisition des images, afin de leur permettre de mieux les appréhender. Le cours balaye les principales modalités d'imagerie médicale et présente l'état de l'art actuel de l'intelligence artificielle en imagerie médicale. Les étudiants sont sensibilisés aux enjeux du domaine et manipulent différentes images dans différents formats.

Une attention particulière est apportée à la place des femmes dans l'histoire de la médecine et de l'imagerie médicale, ainsi que le côté consommation de ressources, risques sanitaires et pollution.

### Contenu et plan du cours

- Introduction et histoire de la médecine et de l'imagerie médicale
- Les Rayons X
- L'IRM
- Les autres techniques d'imagerie
- L'intelligence artificielle en imagerie médicale

### Format des activités

Formats des activités pédagogiques :

- Cours magistraux
- Travaux pratiques

- Projet

### **Acquis d'apprentissage attendus**

A l'issue de ce cours, les étudiants sont capables :

- De connaître les différentes modalités d'imagerie médicale
- De manipuler des formats d'images médicales divers
- D'être conscients des enjeux de l'imagerie médicale

### **Évaluation des acquis d'apprentissage attendus**

- Présentation et critique d'un article issu d'un concours organisé par une conférence en imagerie médicale afin de voir des cas concrets et d'être capable d'avoir un esprit critique dessus.

### **Formule de calcul de la note finale**

Présentation de l'article (grille critérée)

### **Références et bibliographie**

- *Actes de conférences MICCAI*
- *Supports Moodle*

## COFE – Code Fest Week

Niveau : M1	Semestre : S8	Coefficient : <b>1</b>
Volume horaire total : 40	Volume horaire face-à-face : 18	0/0/18/0 (CM/TD/TP/Examen)
Langues : Français	Enseignant référent : Joseph CHAZALON	

### Prérequis

- Techniques de conception
- Techniques de gestion de projet
- Programmation C++ et Python
- Bases de la vision par ordinateur

### Résumé du cours

Au cours d'une semaine de travail collectif, par groupes fluides de 8 personnes (environ 5 groupes), en étroite collaboration avec les enseignants, les étudiants réalisent une expérience innovante et visuelle en réalité augmentée.

Un dispositif matériel est augmenté de capteurs (profondeur, caméra...) et d'effecteurs (projecteurs, lampes...) permettant d'ajouter des interactions et effets nouveaux.

Les enseignants travaillent avec les étudiants pour les aider à organiser leur travail, le réaliser, collaborer et communiquer. Ils peuvent prendre des rôles de : project manager, product owner, stakeholder, coach, developer.

Résolument ludique, cette activité permet aux étudiants de considérer un projet à 360 degrés, dans un cadre très libre, avec la possibilité de créer collectivement un élément de portfolio différenciant.

### Contenu et plan du cours

Cette activité pédagogique se déroule sur une semaine complète, selon l'organisation suivante :

- lundi
  - lancement du projet
  - prise en main de l'existant
  - mise en place de l'infrastructure de collaboration
  - début du travail sur une proposition d'innovation
  - rétrospective sur les différents plans en fin de journée : projet, groupe, personnel
- mardi
  - réunion de démarrage en début de journée

- développement
- premiers tests et collecte de retours utilisateurs
- préparation de la démonstration du MVP
- rétrospective
- mercredi
- démonstration des MVPs
- identification de synergies possibles entre les groupes
- travail sur une première fonctionnalité innovante
- rétrospective
- jeudi
- réunion de démarrage en début de journée
- développement
- tests et collecte de retours utilisateurs
- préparation de la démonstration finale
- rétrospective
- vendredi
- réunion de démarrage en début de journée
- finalisation des travaux (version stable), tentative d'une 2e fonctionnalité innovante si possible (instable)
- préparation des démonstrations finales
- démonstrations et archivage des travaux
- bilan et rétrospective finale

### **Format des activités**

- Format 100% atelier sur une semaine. 9h-18h tous les jours.
- Alternance de moments de briefing et de moments de travail collectif.
- Travail à la maison à la discrétion des étudiants, facultatif.

### **Acquis de formation attendus**

Garantir son engagement personnel au sein du projet en identifiant ses compétences, envies, besoins, aspirations.

- Je demande de l'aide quand j'en ai besoin
- Je réponds aux demandes d'aide de mes collaborateurs
- Je connais mes forces/mes faiblesses
- Je trouve un équilibre entre ce que je veux faire/je dois faire
- Je tiens mes deadlines, je préviens mes camarades quand ça dépasse.

Définir une solution qui répond à un problème identifié et reposant sur une recherche approfondie de l'existant.

- Les codes/ressources existantes sont utilisés (compréhension...)
- Avoir compris qu'on cherche qqch de ludique/qui en jette/qui capte à destination de prospect
- Une définition de l'utilisateur et de son parcours (qu'est-ce que souhaite le prospect).

Collaborer à la construction d'une réponse collective d'une solution contrainte dans un environnement spécifié. Organisation du travail incrémentale et itérative.

- qui est responsable de quoi
- mettre en place un référentiel unique d'info
- mettre en place des outils de comm
- mettre en place VCS / outillage / CI/CD...
- mettre en place métho de décision collective basée sur des éléments objectifs et mesurables
- les tâches à réaliser sont identifiées et réparties
- des check-ups réguliers de l'avancement (Daily sont respectés)

Livrer une réponse qualifiée à la demande initiale qui s'appuie sur des choix argumentés et une démonstration exportable dans différent contexte, sous forme de MVP.

- Avoir un MVP démontrable/fonctionnel
- Avoir un retour utilisateur
- mettre en place une méthodologie de test avec plusieurs niveaux de maturité techno (environnement plus ou moins simulé/réaliste)
- itérer sur la solution au regard des tests/retours d'usage/imprévus/pivots/choix
- démontrer ses choix

### Évaluation des acquis de formation attendus

Un certain nombre de livrables destinés à l'accompagnement des étudiants sont à compléter au cours du projet.

Pour chacun d'entre eux, un retour rapide (non réalisé, réalisation superficielle, réalisation rigoureuse) sera fourni (soit par les enseignants, soit par les étudiants entre eux).

La quantité de livrables produits de façon rigoureuse sera utilisée pour calculer une indication finale de la contribution à l'effort collectif.

La pertinence des solutions proposées sera évaluée de façon croisée.

#### **Formule de calcul de la note finale**

Moyenne des évaluations individuelles pour chacun des livrables.

#### **Référence et Bibliographie**

À compléter.

## IRGPU - Implémentation Rapide GPU

Niveau : M1/M2	Semestre : Automne	Coefficient :
Volume horaire total : 30	Volume horaire face-à-face : 14	8 / / 6 / 1 (CM/TD/TP/Examen)
Langues :	Enseignant référent : Edwin Carlinet* & Joseph Chazalon	

### Prérequis

- Niveau intermédiaire en C++
- Des notions de programmation parallèle sont souhaitées



### Résumé et contexte

Les évolutions en matière d'architecture des processeurs tendent à pousser vers une généralisation des architectures multicœurs et massivement parallèle tels que les GPUs. Dans ce contexte, la maîtrise des techniques de programmation massivement parallèle est devenue obligatoire pour tirer parti des performances de ces nouveaux systèmes. L'objectif de ce cours est de poser les bases nécessaires à la programmation de GPUs avec CUDA et d'aborder les motifs récurrents d'optimisation GPU illustrés sur des problèmes de traitement d'images.



### Acquis d'apprentissages visés (AAVs)

À l'issue de ce cours, les étudiants sont capables dans le cadre du développement d'une application exploitant les capacités d'une architecture massivement parallèle de :

- Comprendre les différences fondamentales entre des architectures CPUs et GPUs et leurs implications sur la programmation
- Mettre en oeuvre des benchmarks de performance et identifier les parties critiques d'un programme en termes de performances.
- Identifier les parties massivement parallélisables d'un programme et leurs motifs GPU correspondant.
- Déporter des parties critiques d'un programme sur GPU en utilisant CUDA.
- Analyser les performances d'un kernel GPU et identifier ses goulots d'étranglement
- Optimiser un kernel GPU en appliquant les techniques d'optimisations standard vues en cours.



### Plan cours

Le cours est organisé sur deux axes : fondements théoriques et applications à la programmation.

#### Chapitre 1 : Introduction aux modèles de programmation parallèle et massivement parallèle

Après un rapide historique de l'évolution des cartes graphiques en tant qu'accélérateur de calcul, l'étudiant verra les différents modèles le parallélisme et leur correspondance hardware. En particulier, il verra le modèle de programmation CUDA pour les architectures GPU NVidia.

## Chapitre 2 : Introduction pratique à CUDA (2h CM + 2h TP)

Après la vision formelle du modèle de programmation CUDA, l'étudiant aura une formation pratique à CUDA au cours de laquelle il verra comment lancer des kernels pour paralléliser des calculs simples sur des données 1D et 2D.

## Chapitre 3 : Programmation efficace des GPUs (1e partie)

L'étudiant verra comment paralléliser des motifs simples pour résoudre des problèmes récurrents (illustrés par des problèmes de traitement d'images). Pendant cette visite, il prendra conscience des problèmes potentiels de performances (notamment liées à la mémoire) et comment les résoudre.

Il s'ensuit un TP d'application où l'étudiant doit paralléliser le calcul d'une synthèse d'image avec CUDA.

## Chapitre 4 : Programmation efficace des GPUs (2e partie)

Cette deuxième partie se concentre sur les motifs de réductions dont la parallélisation est moins triviale.

L'étudiant devra ensuite appliquer ces connaissances dans le cadre d'un exercice de réduction du TP précédent.



## Format des activités

Formats des activités pédagogiques:

- Cours magistraux: 8 heures
- Travaux pratiques: 6 heures



## Évaluation des AAVs

3 QCMs

Projet



## Calcul de note finale

- Contrôle continu (QCM): 3 points
- Projet & soutenance: 17 points



## Références et bibliographie

Kirk, D. B., & Wen-Mei, W. H. (2016). *Programming massively parallel processors: a hands-on approach*. Morgan kaufmann.